Sketching Data With T-Digest

Erik Erlandson

Red Hat, Inc.

email: eje@redhat.com
twitter: @manyangled
github: erikerlandson

Why Sketching?

Why Sketching?

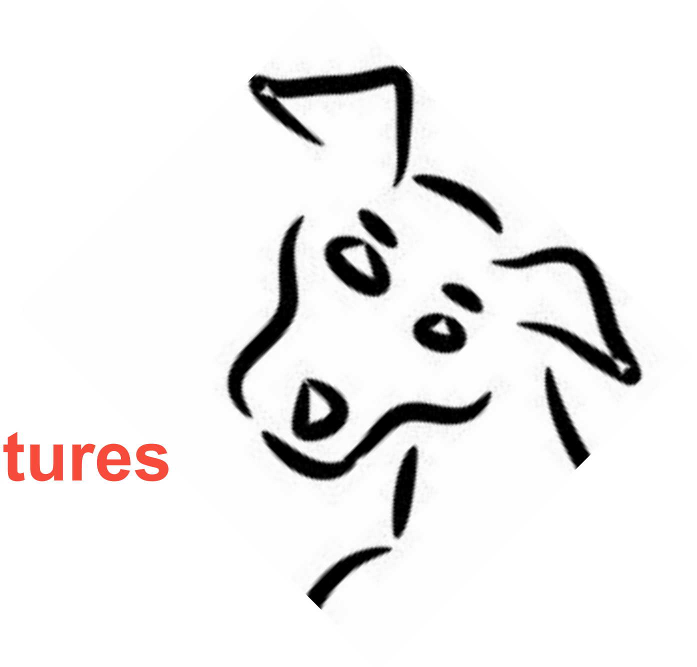
- Smaller

Why Sketching?

- Smaller
- Faster

Why Sketching?

- Smaller
- Faster
- Essential Features



We All Sketch Data

3.4

6.0 → Mean = 3.97

Variance = 3.30

2.5

⋮

We All Sketch Data

3.4

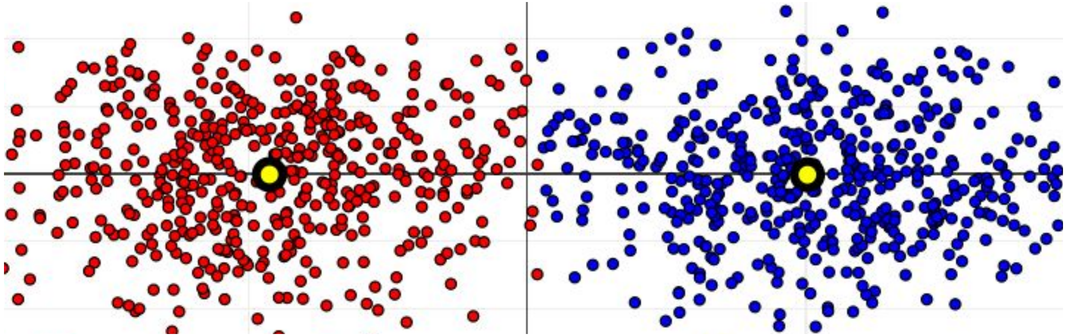
6.0

2.5

⋮

Mean = 3.97

Variance = 3.30



We All Sketch Data

3.4

6.0

2.5

⋮

Mean = 3.97

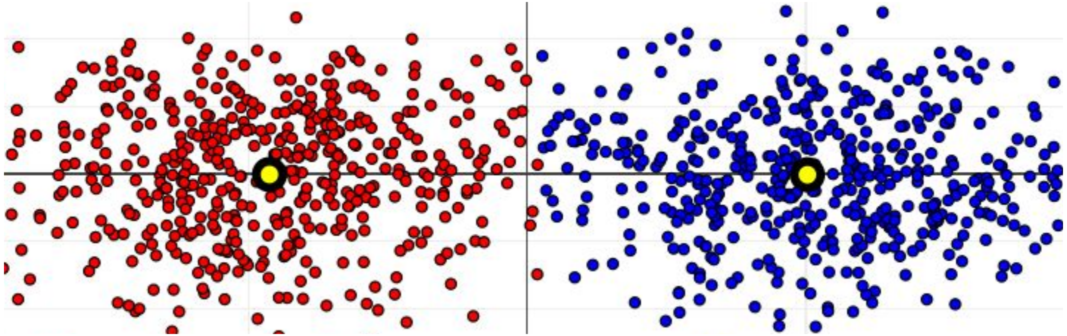
Variance = 3.30

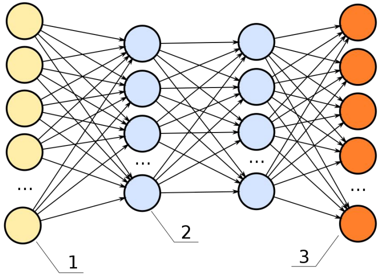
3.4, 5.0, 9.0

6.0, 2.1, 7.7

2.5, 4.4, 3.2

⋮





T-Digest

Computing Extremely Accurate Quantiles Using t-Digests

Ted Dunning & Omar Ertl

Java, Python, R, JS, C++ and Scala

Library for Spark and PySpark

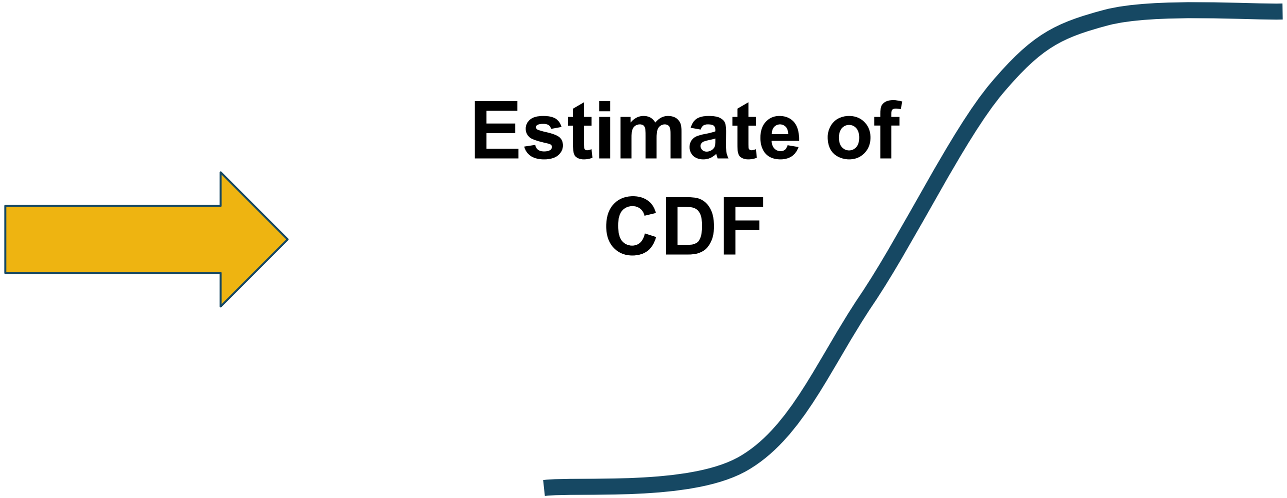
What is T-Digest Sketching?

3.4

6.0

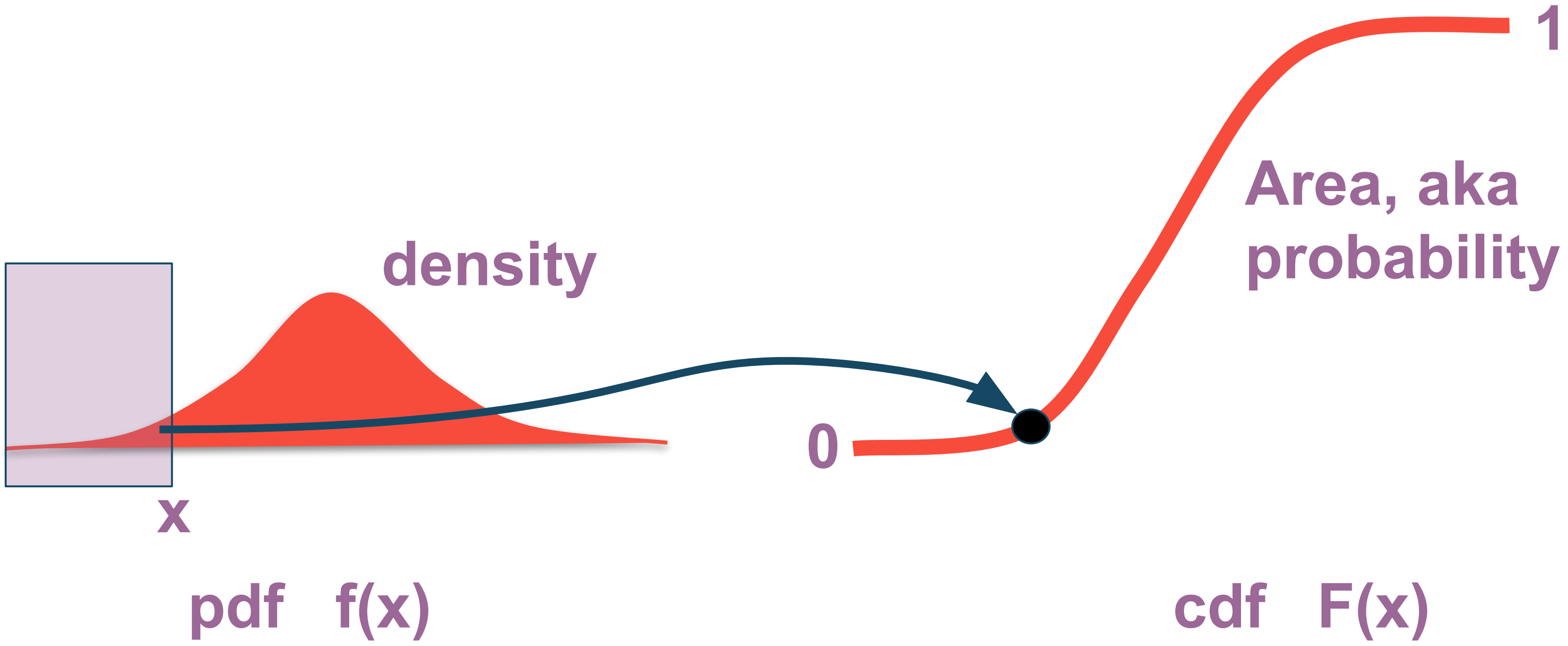
2.5

⋮

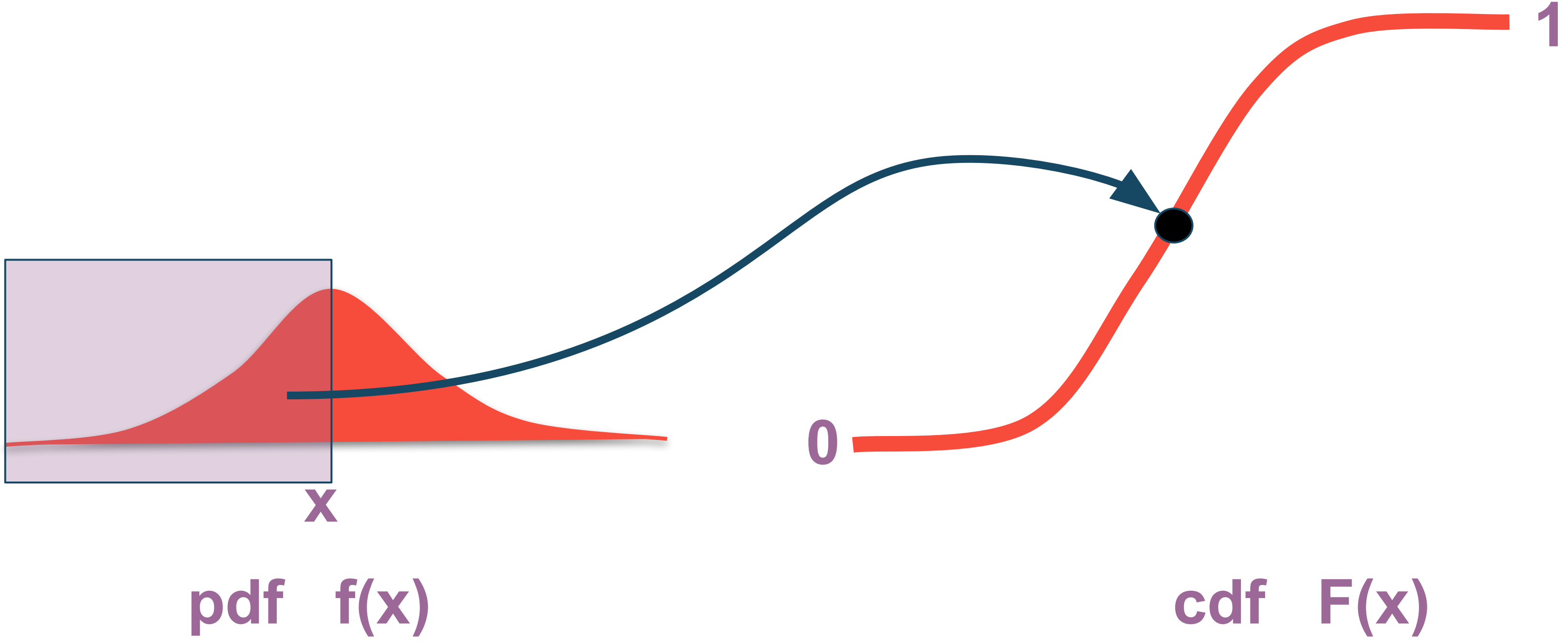


Estimate of CDF

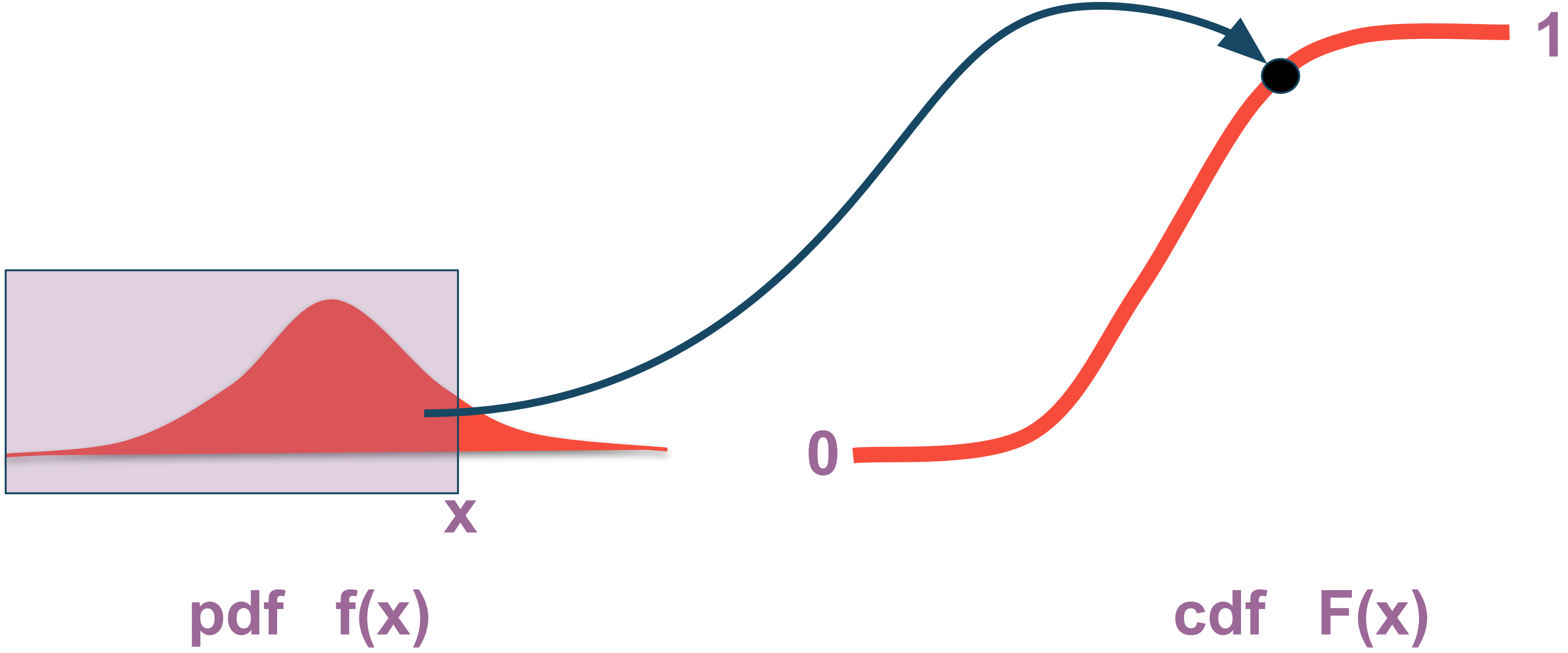
Cumulative Distribution



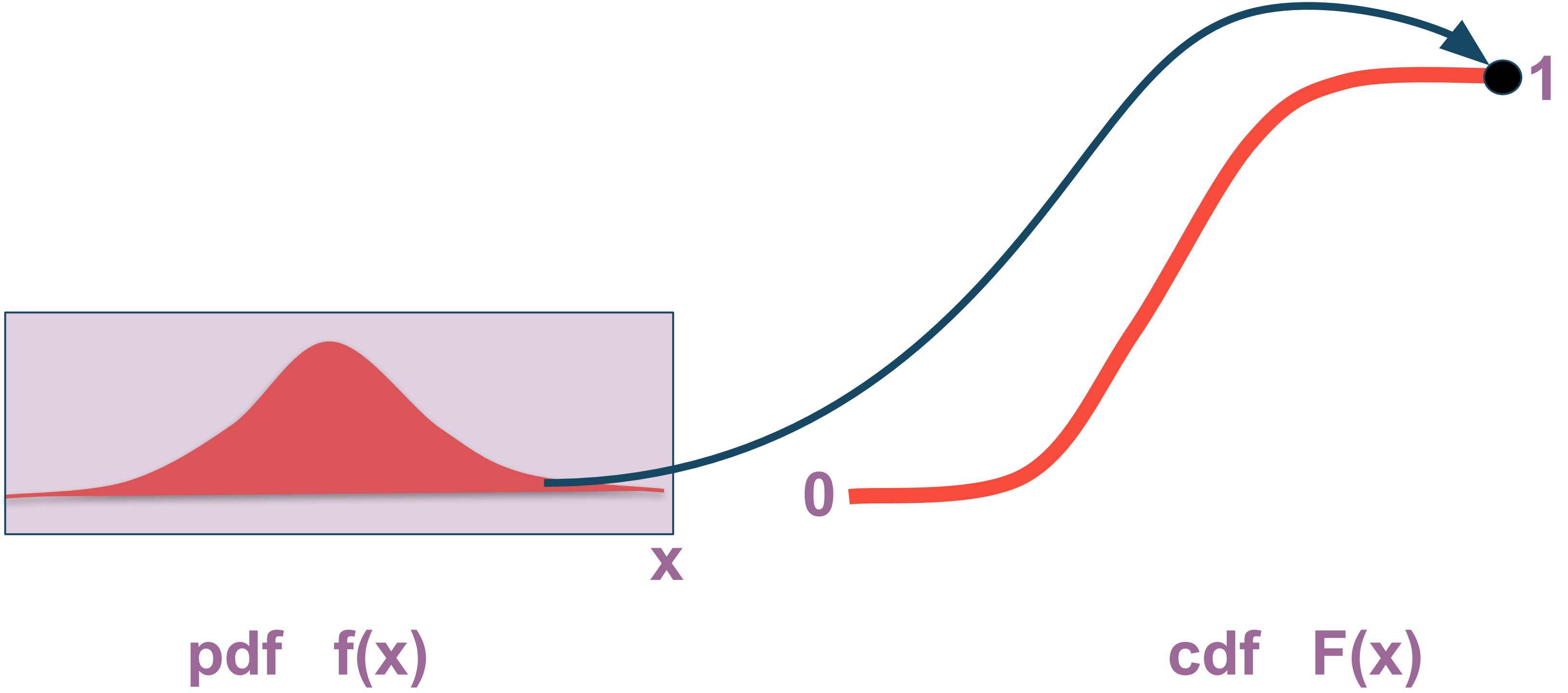
Cumulative Distribution



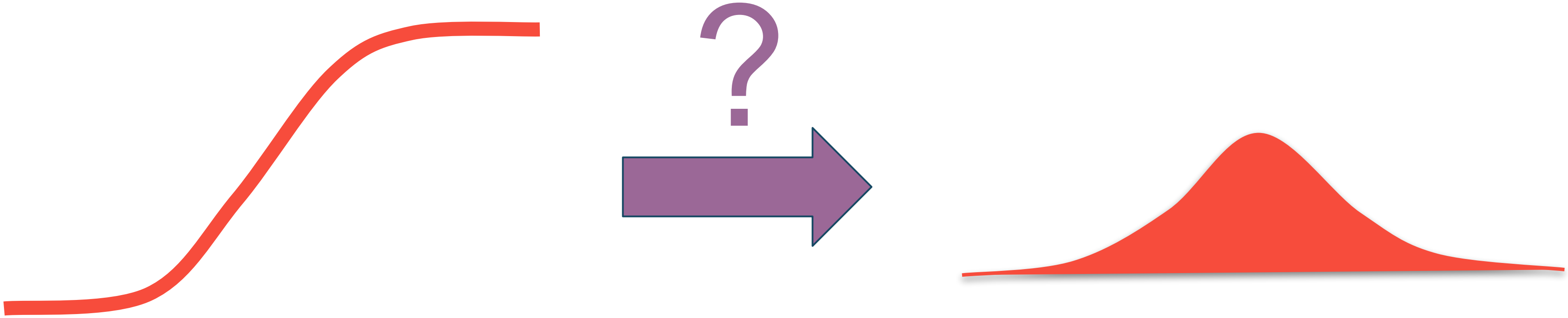
Cumulative Distribution



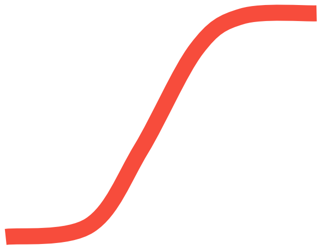
Cumulative Distribution



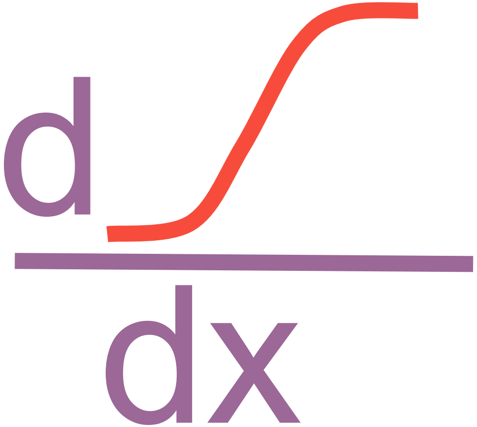
But What About The Density?



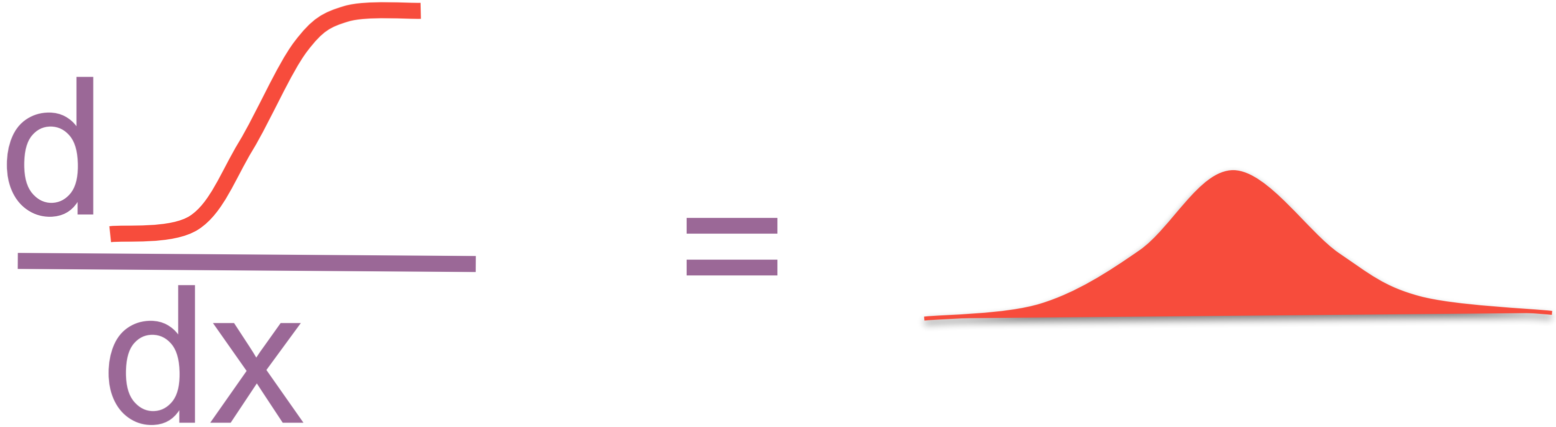
Cumulative Distribution



Cumulative Distribution



Cumulative Distribution



Incremental Updates

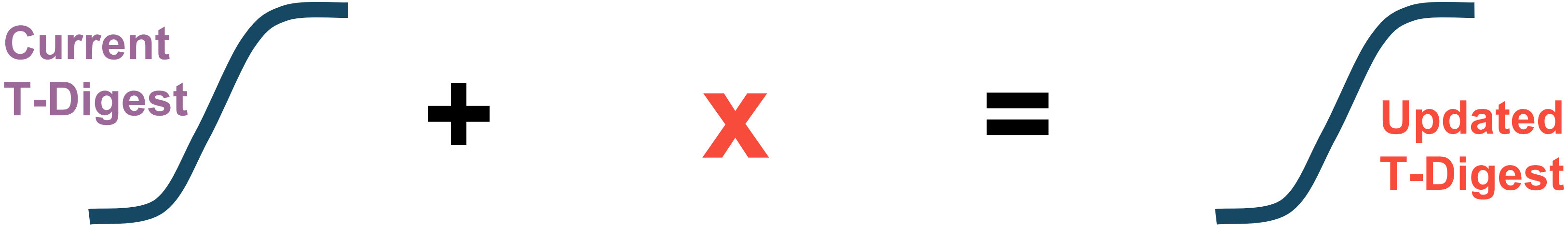
Current T-Digest

Incremental Updates

Current T-Digest

+ x

Incremental Updates



Current T-Digest

+

x

=

Updated T-Digest

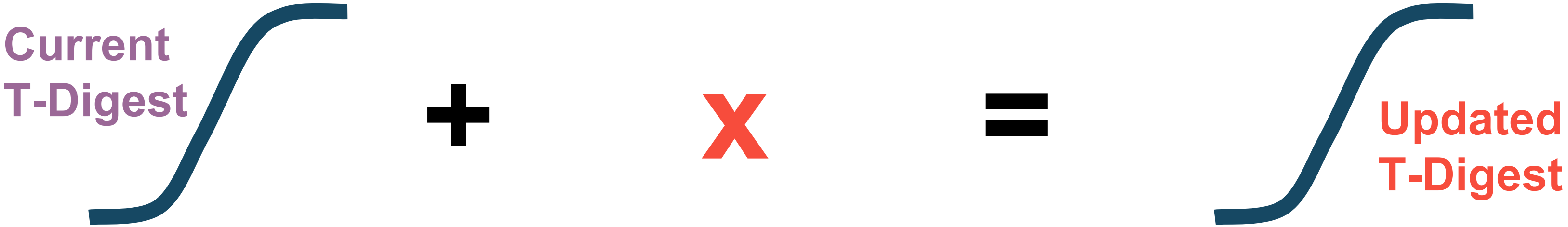
Incremental Updates

Current T-Digest

+ x =

Updated T-Digest

Incremental Updates



Current T-Digest

+

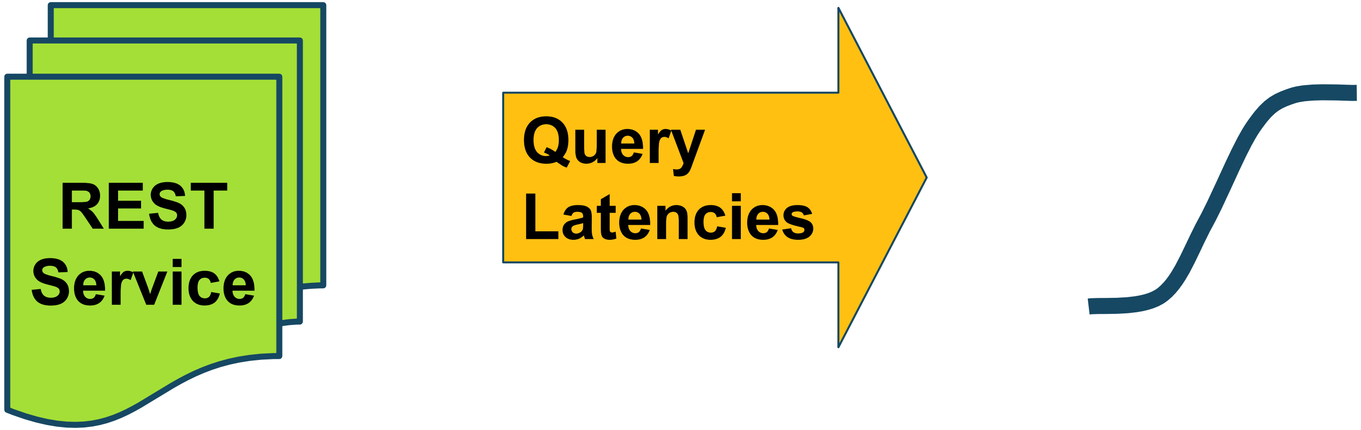
x

=

Updated T-Digest

Compact “Running” Sketch

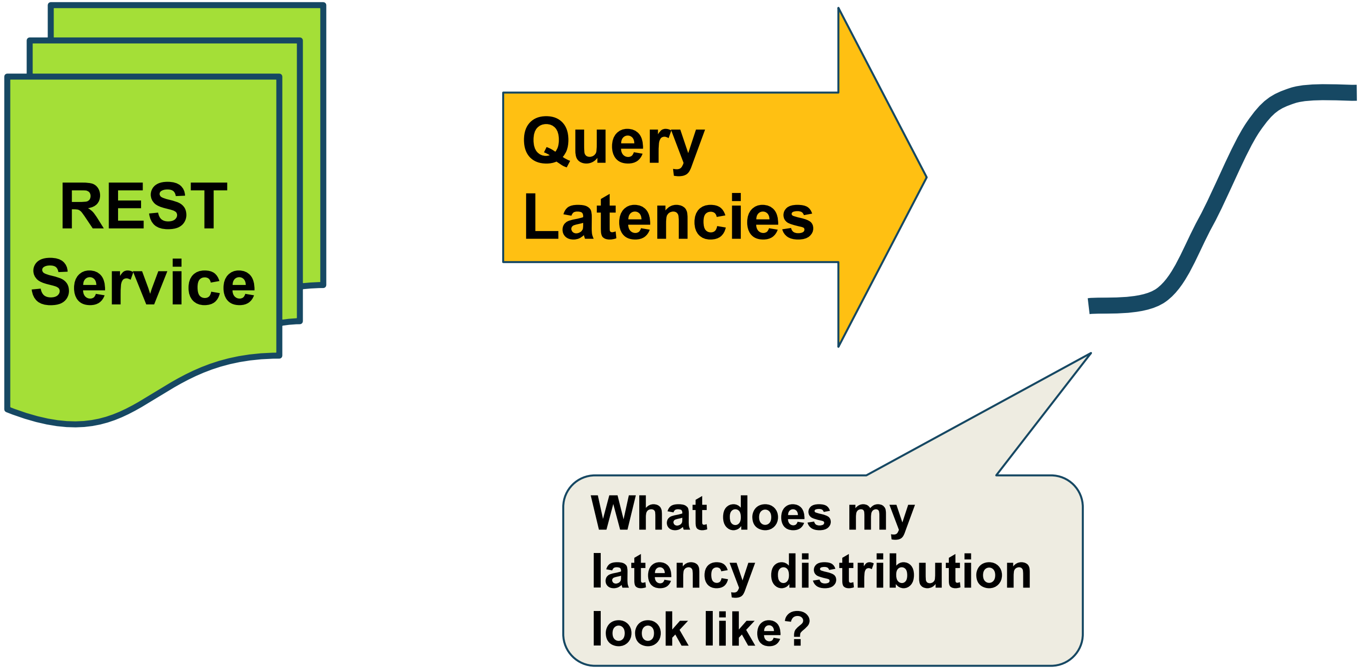
Payoff



REST Service

Query Latencies

Payoff

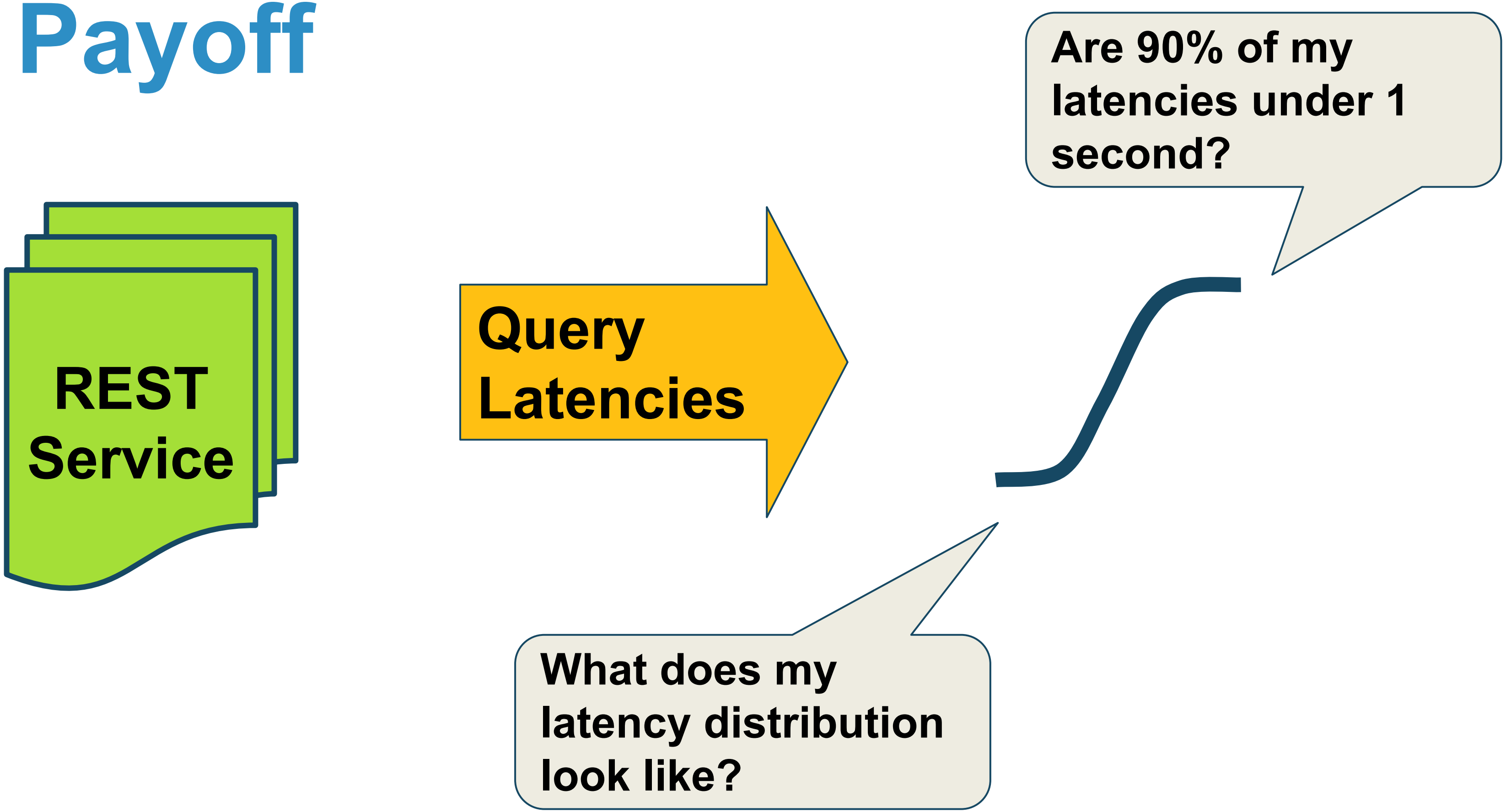


REST Service

Query Latencies

What does my latency distribution look like?

Payoff



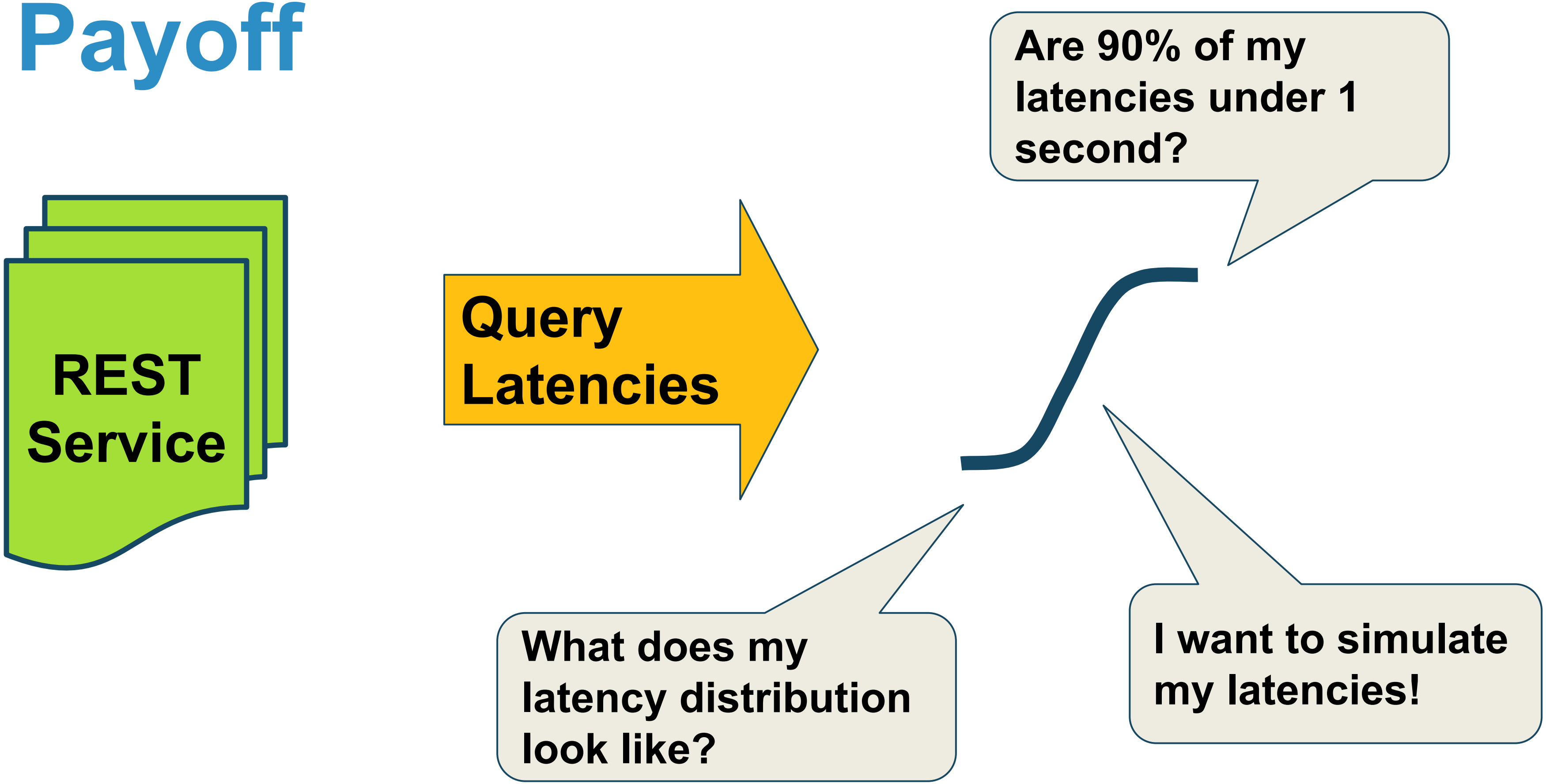
REST Service

Query Latencies

Are 90% of my latencies under 1 second?

What does my latency distribution look like?

Payoff



REST Service

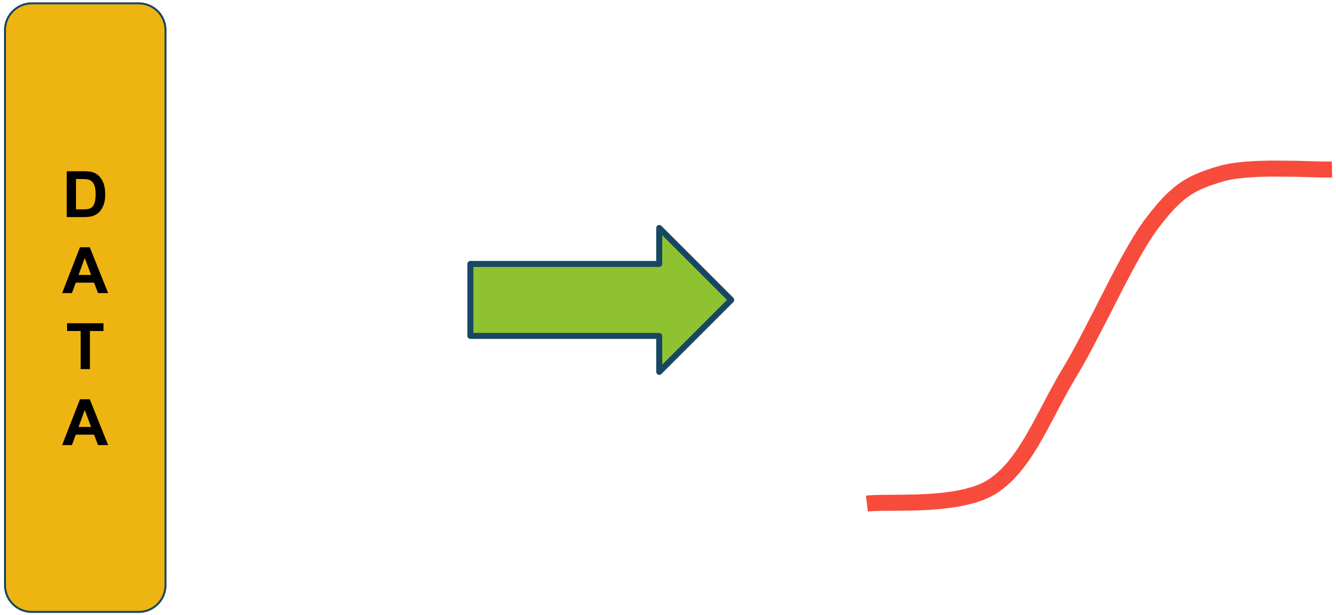
Query Latencies

Are 90% of my latencies under 1 second?

What does my latency distribution look like?

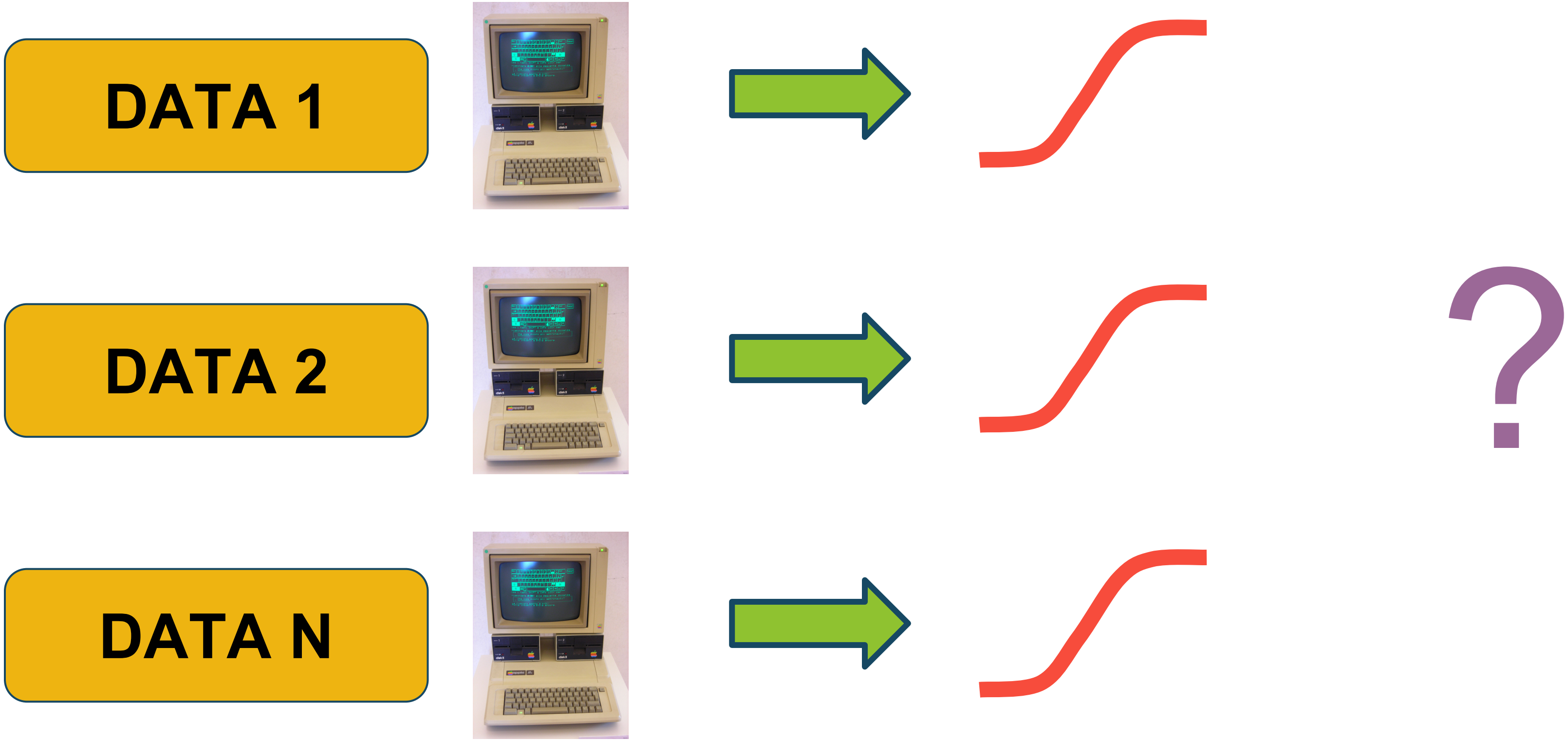
I want to simulate my latencies!

Even More Payoff



D
A
T
A

Even More Payoff

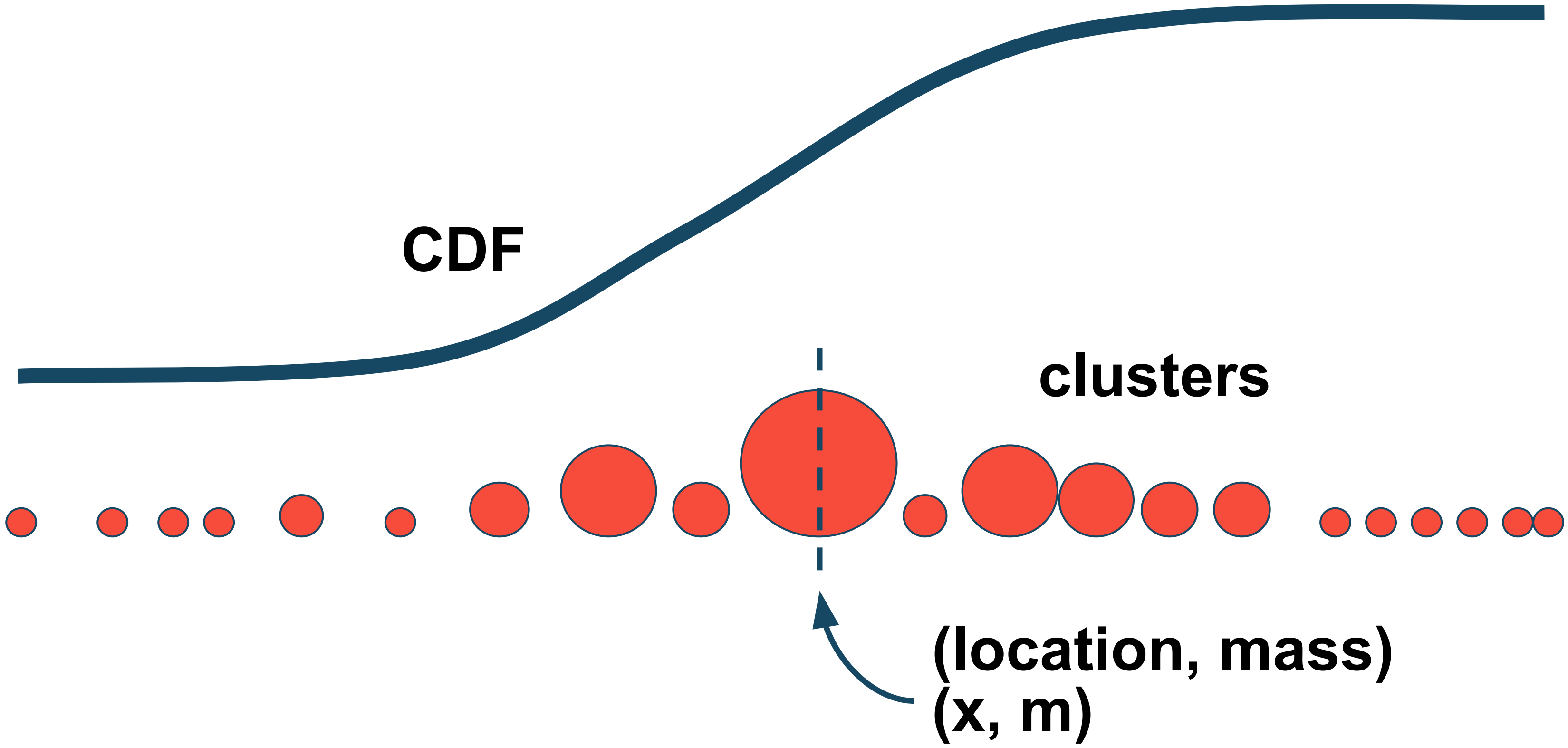


DATA 1

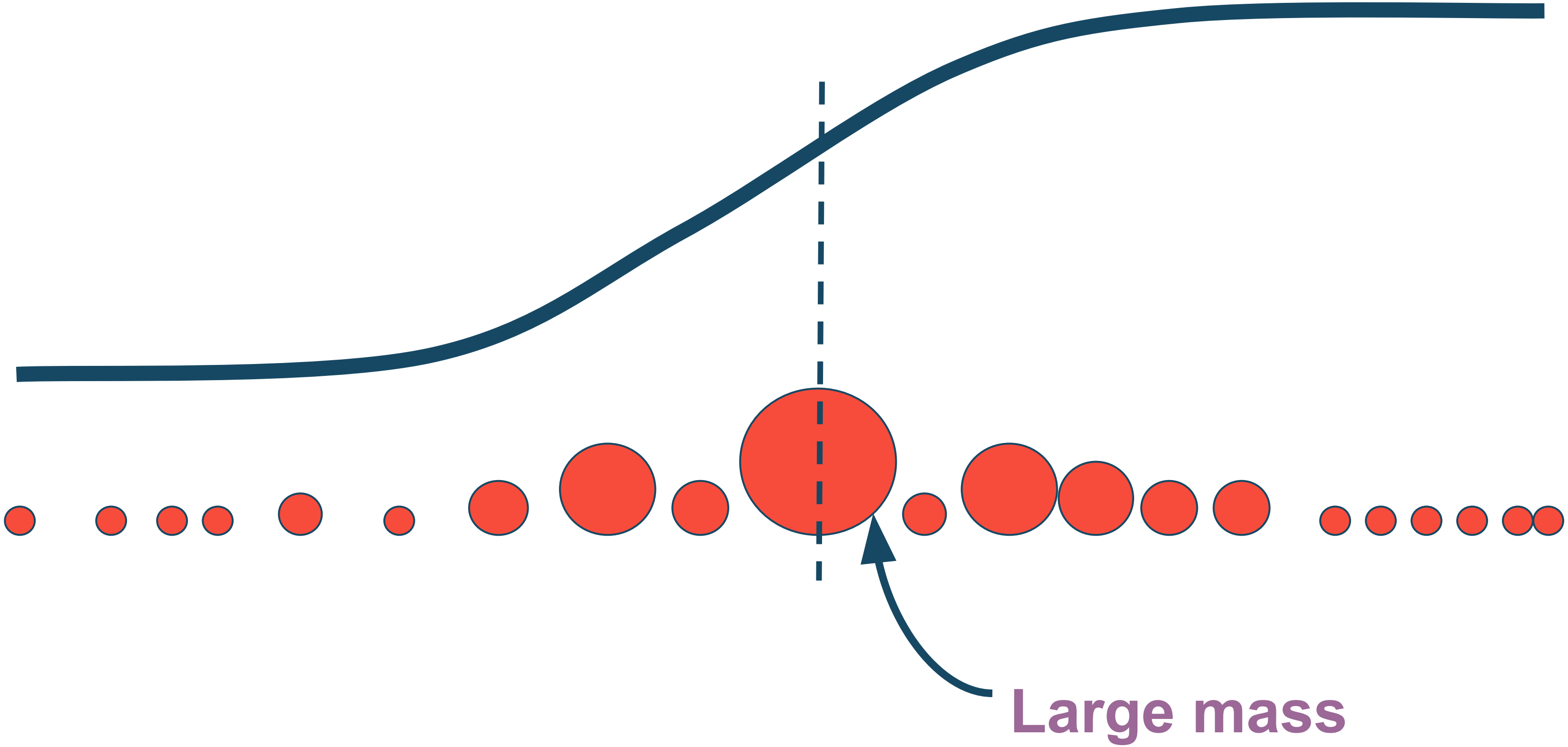
DATA 2

DATA N

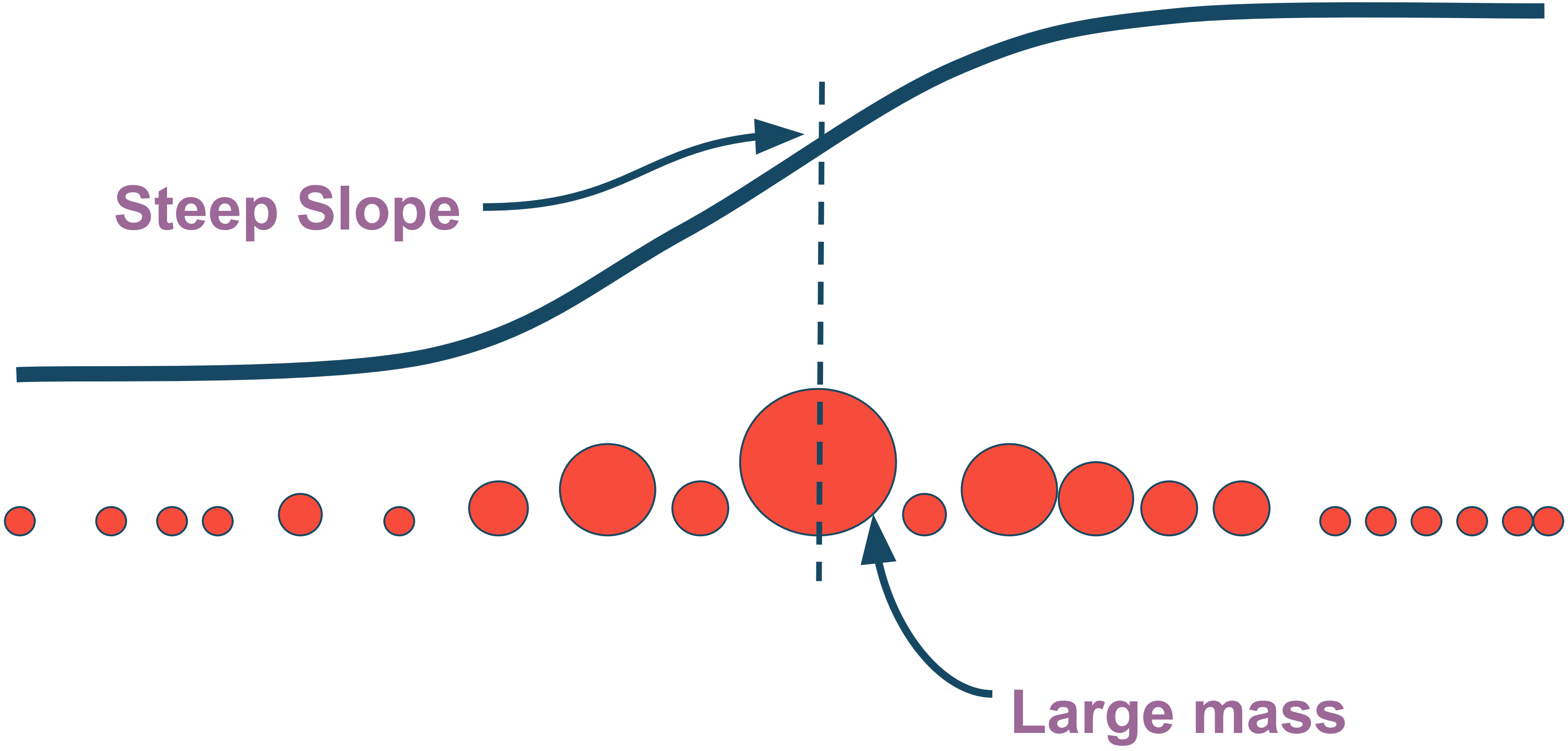
Representation



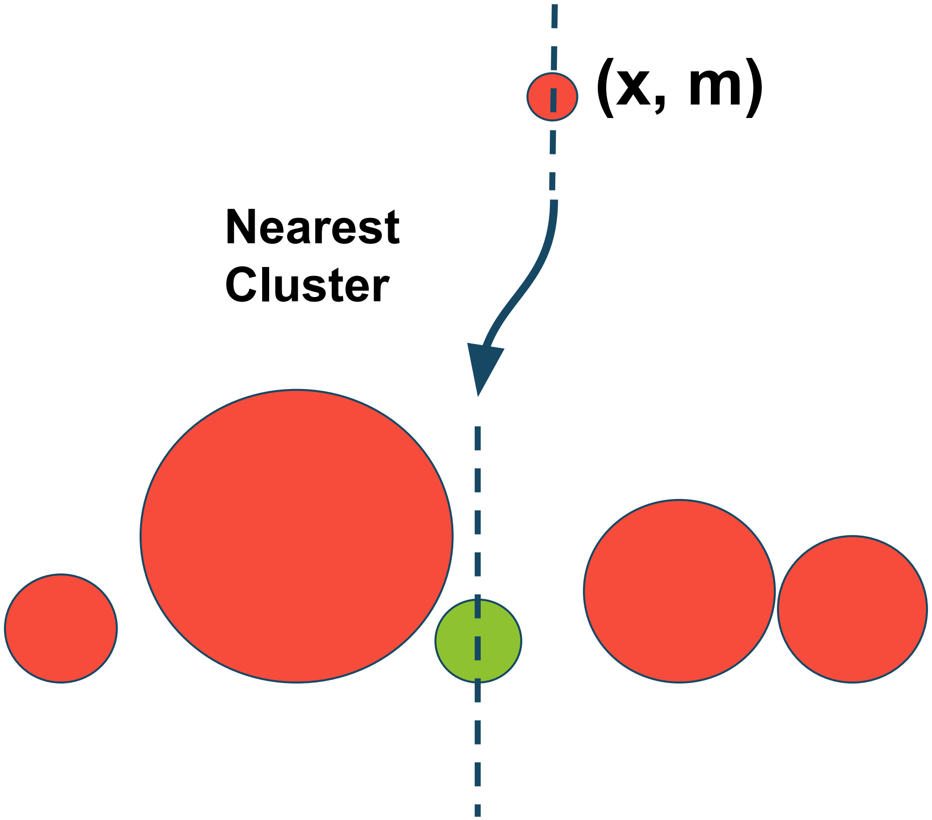
Representation



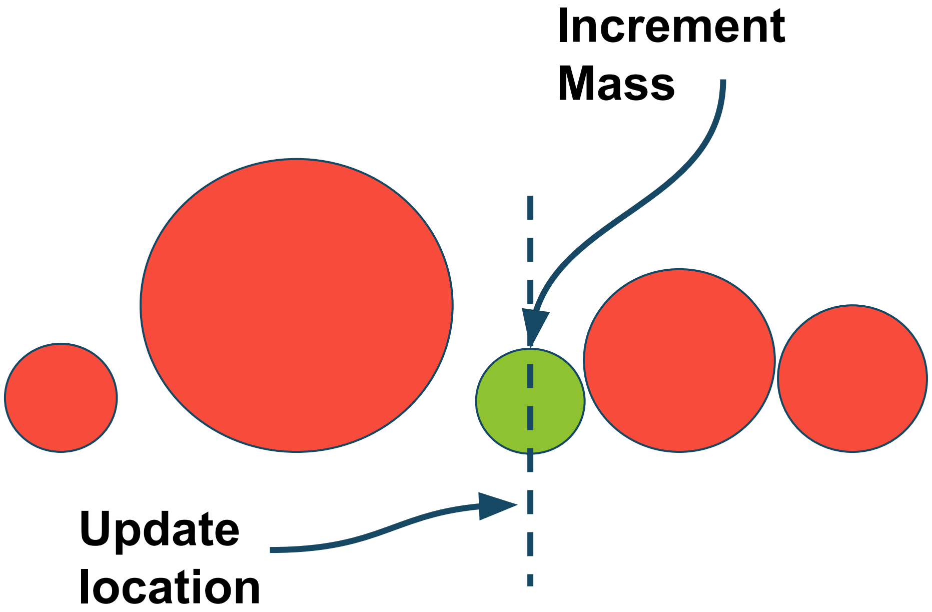
Representation



Update

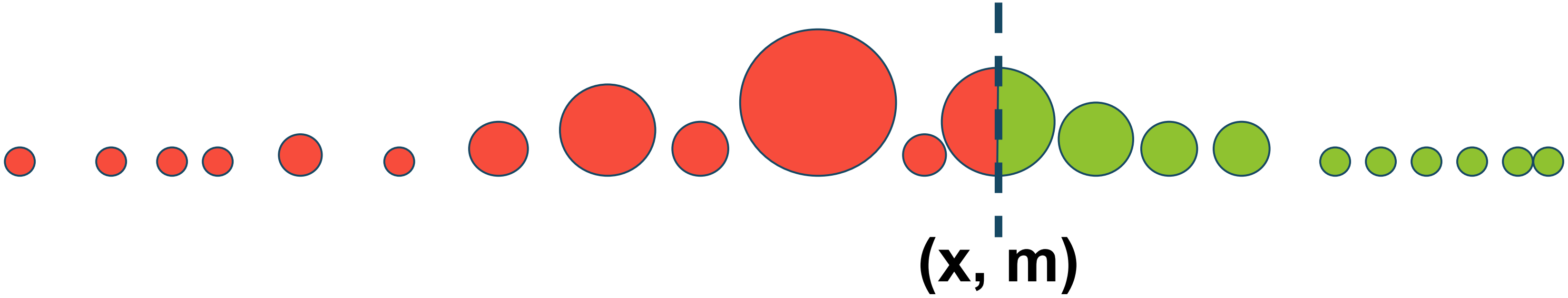


Update

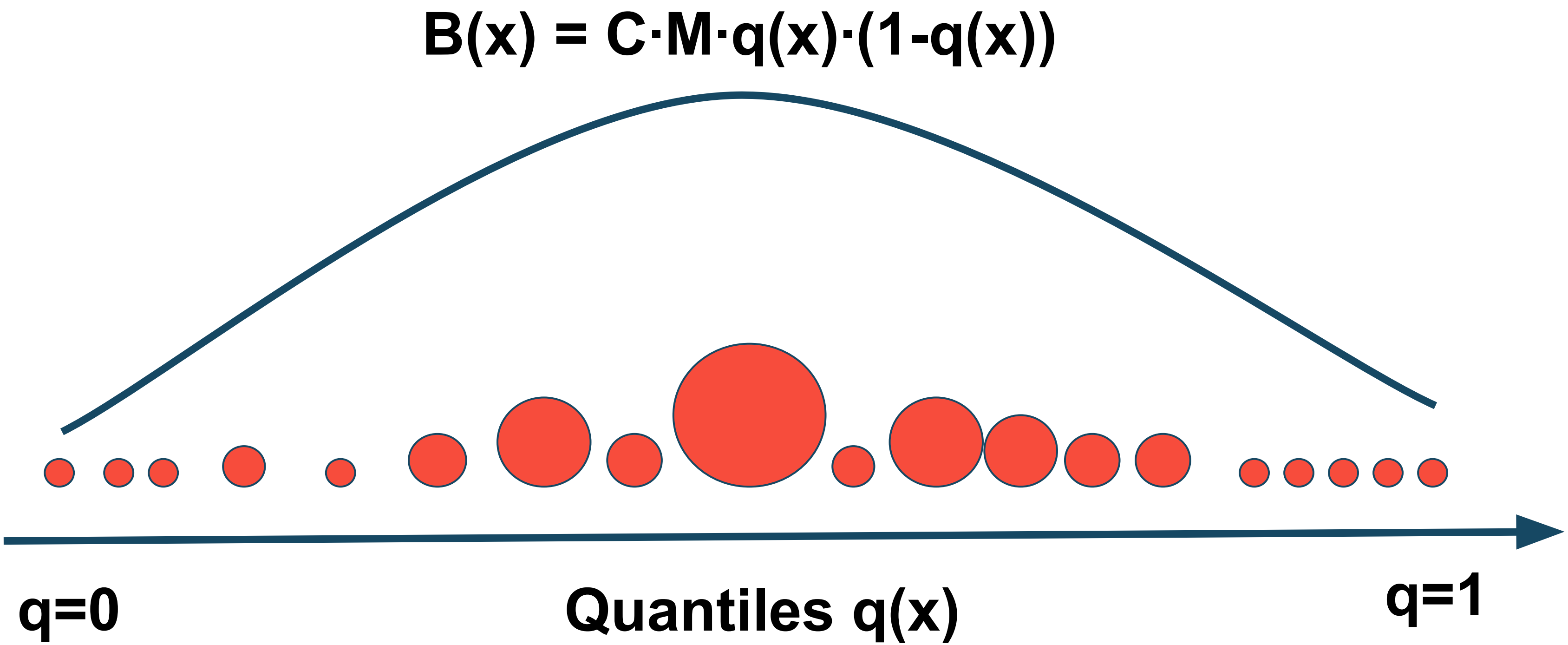


Cluster Quantile

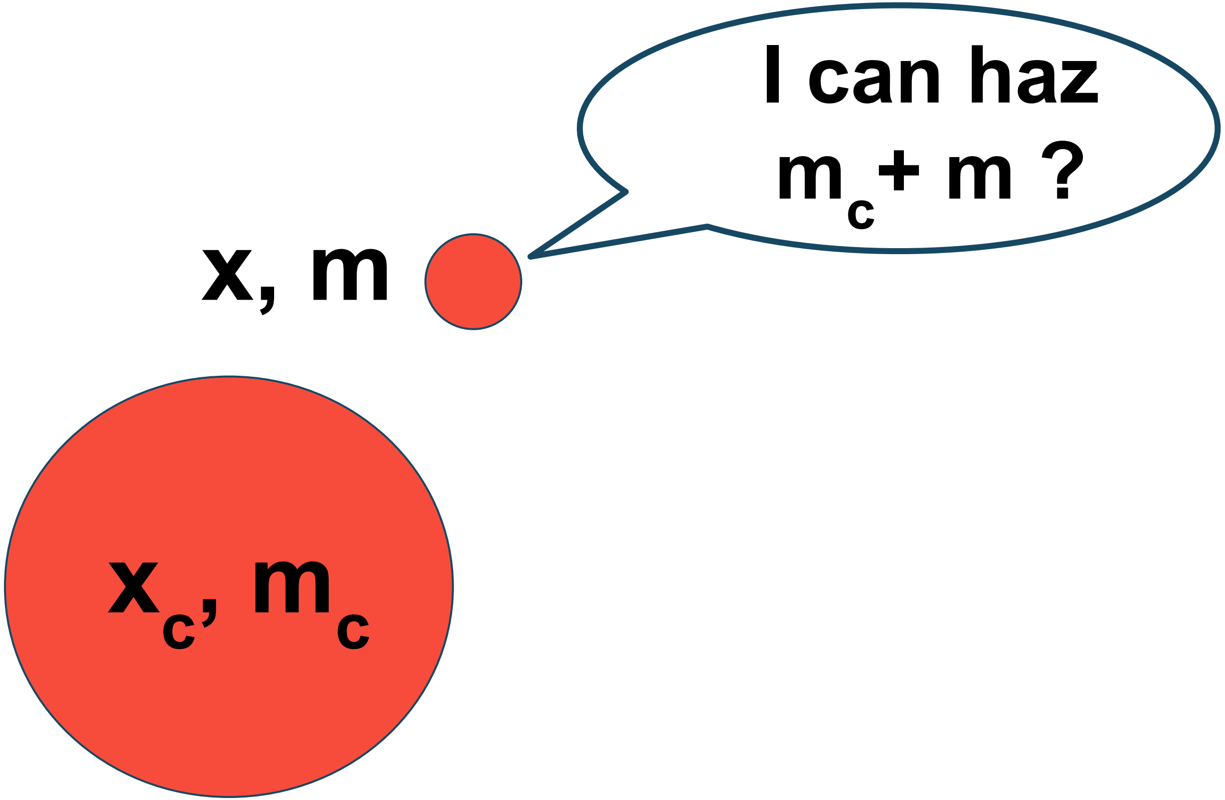
$$q(x) = \frac{\sum \text{🔴}}{\sum \text{🔴} + \sum \text{🟢}}$$



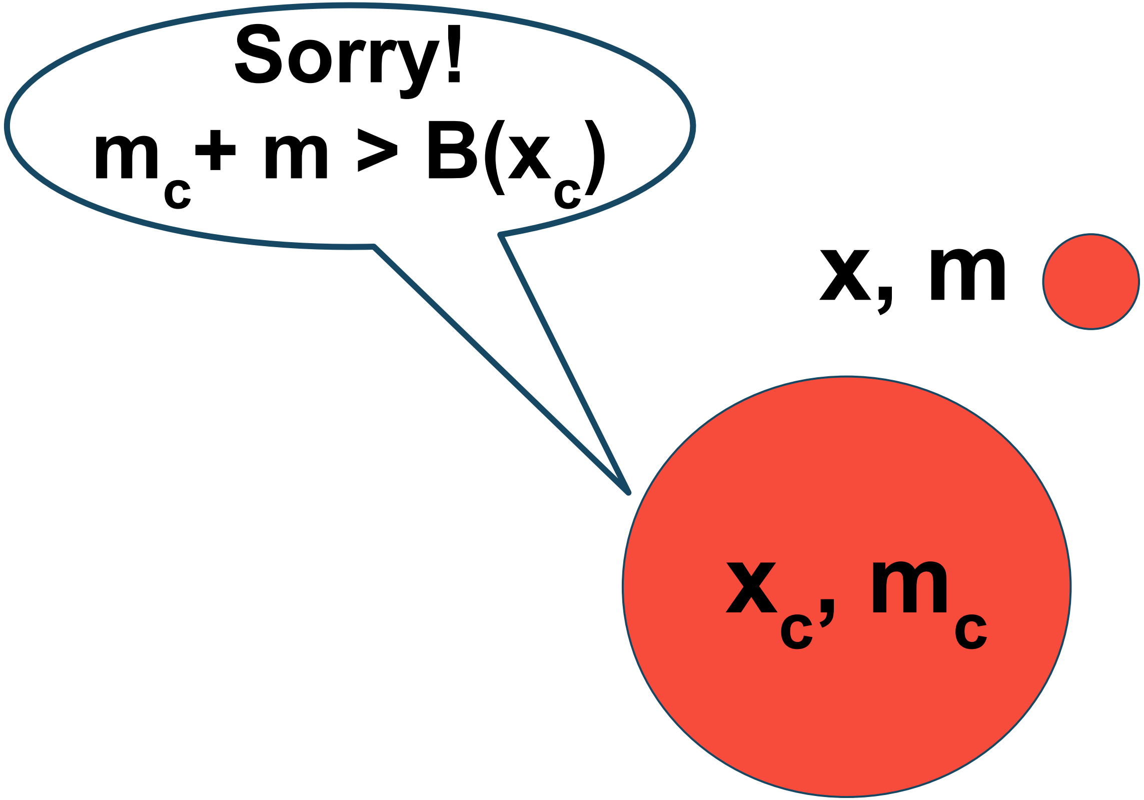
Cluster Mass Bounds



Bounds Force New Clusters



Bounds Force New Clusters



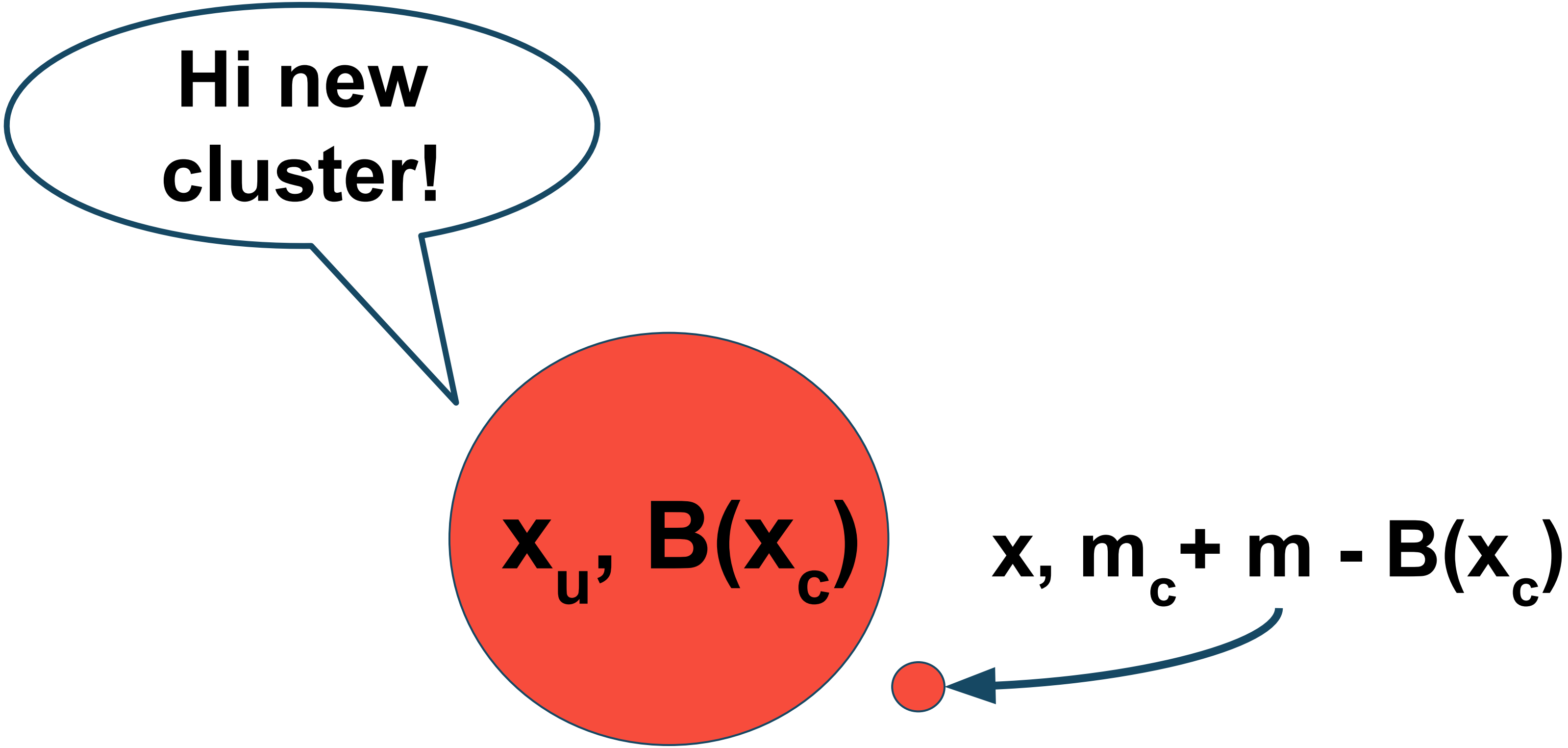
Sorry!

$m_c + m > B(x_c)$

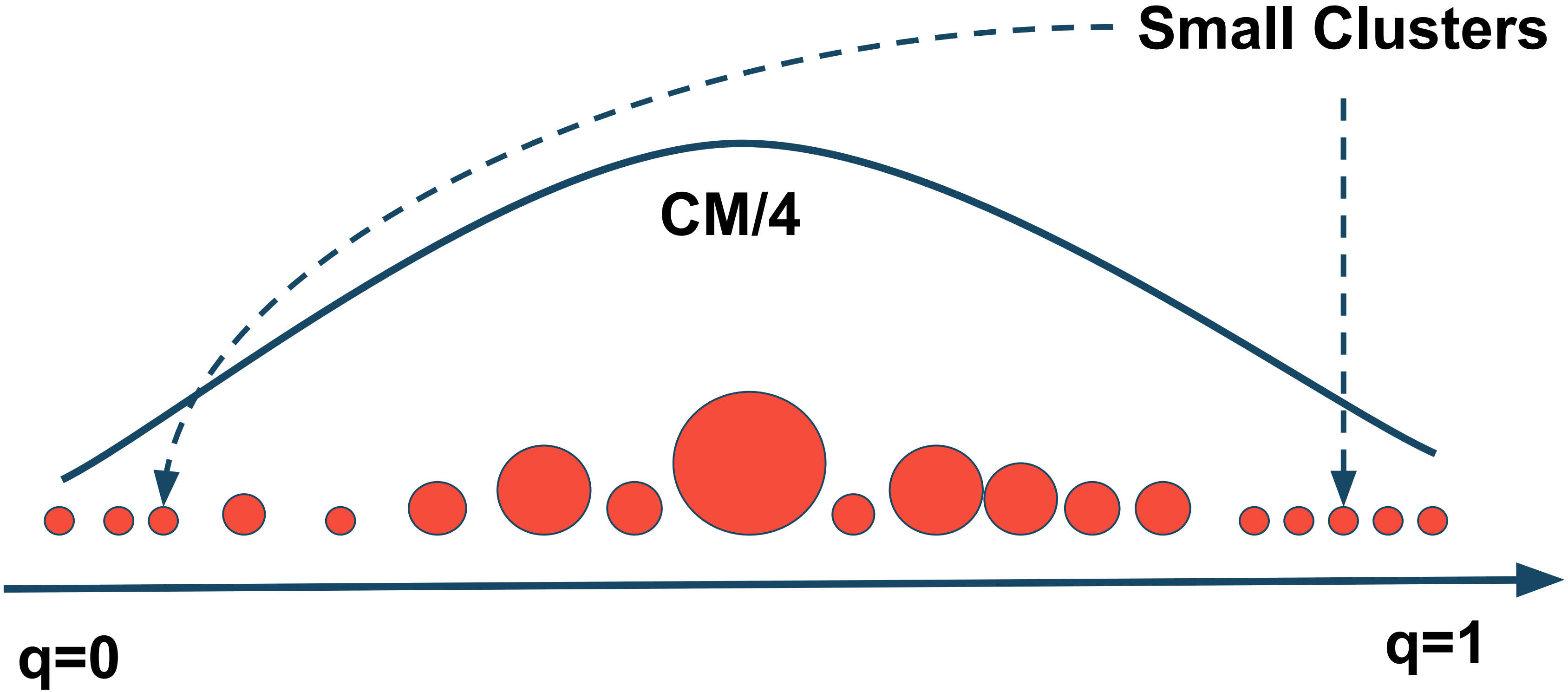
x, m

x_c, m_c

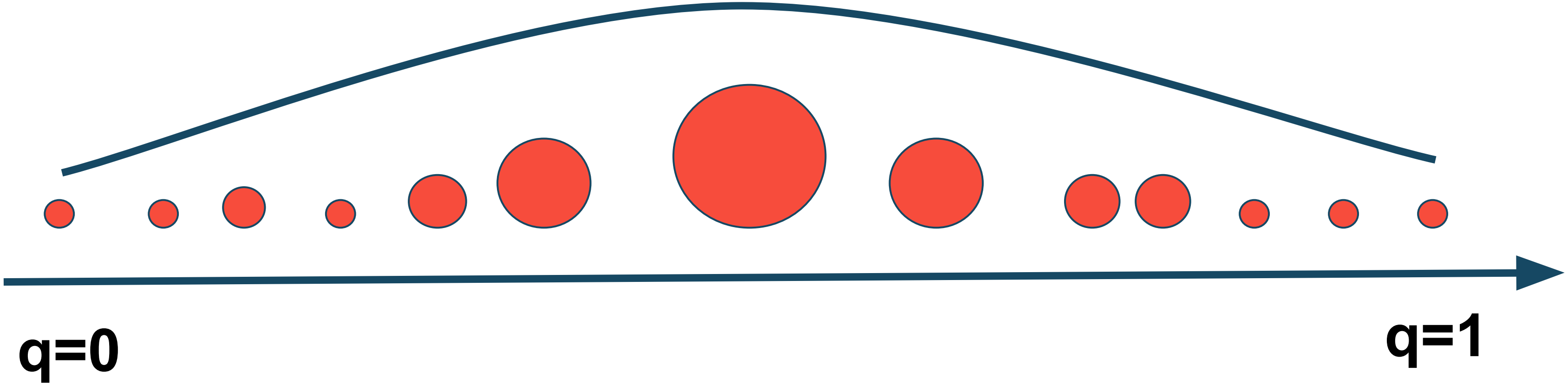
Bounds Force New Clusters



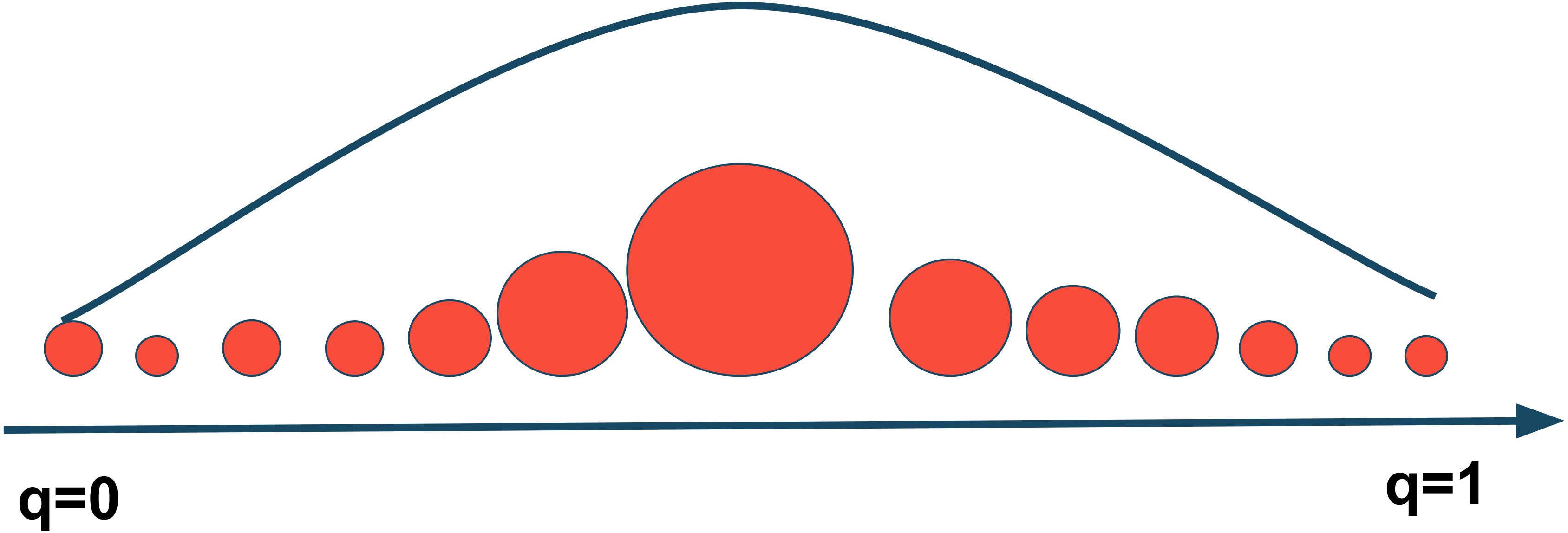
$$B(x) = C \cdot M \cdot \underline{q(x) \cdot (1-q(x))}$$



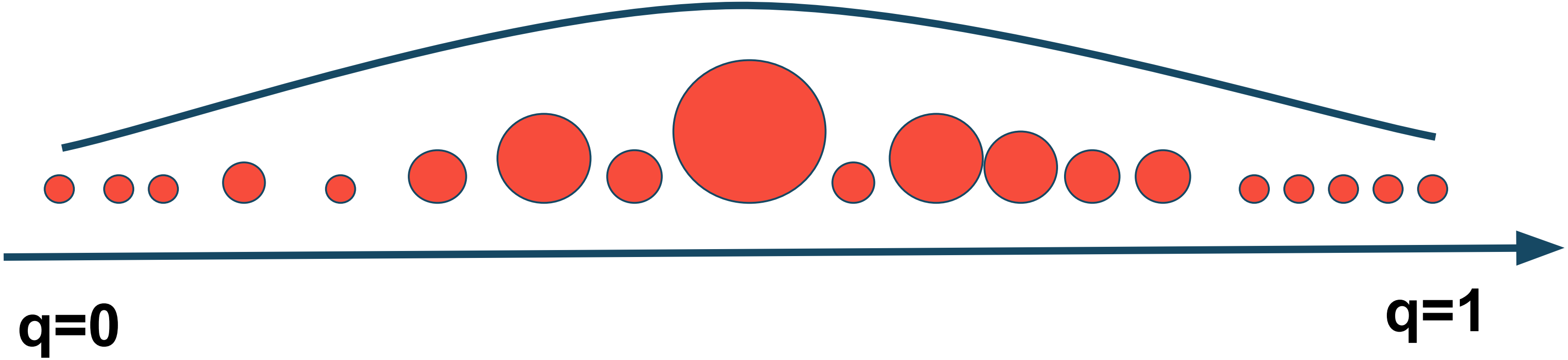
$$B(x) = C\cdot\underline{M}\cdot q(x)\cdot(1-q(x))$$



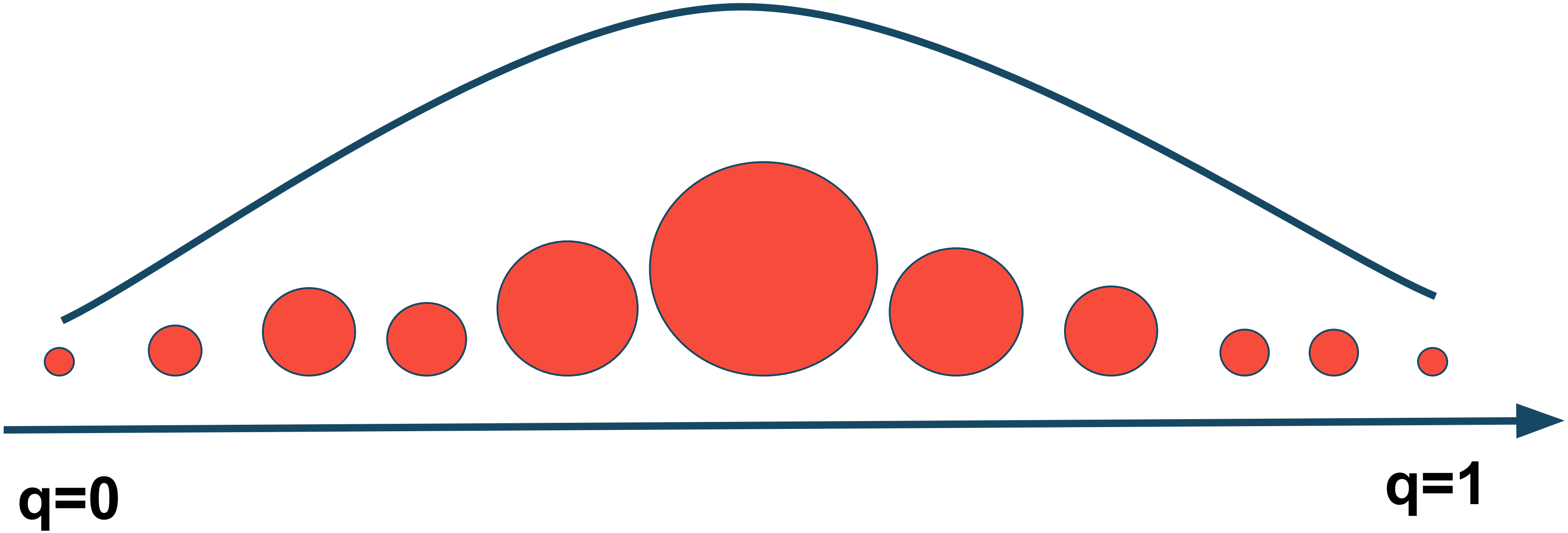
$$B(x) = C \cdot \underline{M} \cdot q(x) \cdot (1-q(x))$$



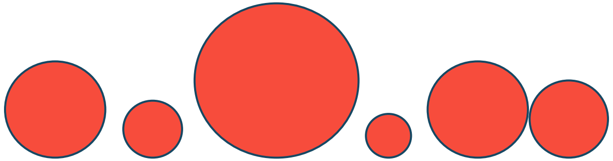
$$B(x) = \underline{C}\cdot M\cdot q(x)\cdot(1-q(x))$$



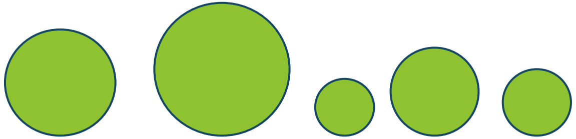
$$B(x) = \underline{C} \cdot M \cdot q(x) \cdot (1 - q(x))$$



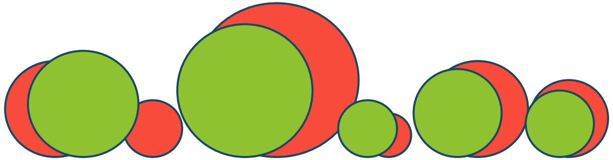
T-Digests are Mergeable



$D_1 \equiv C_1$

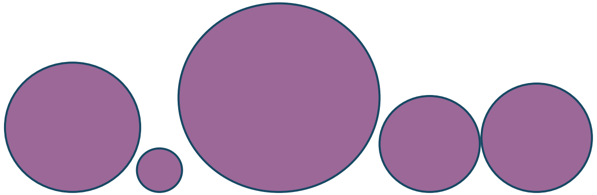


$D_2 \equiv C_2$



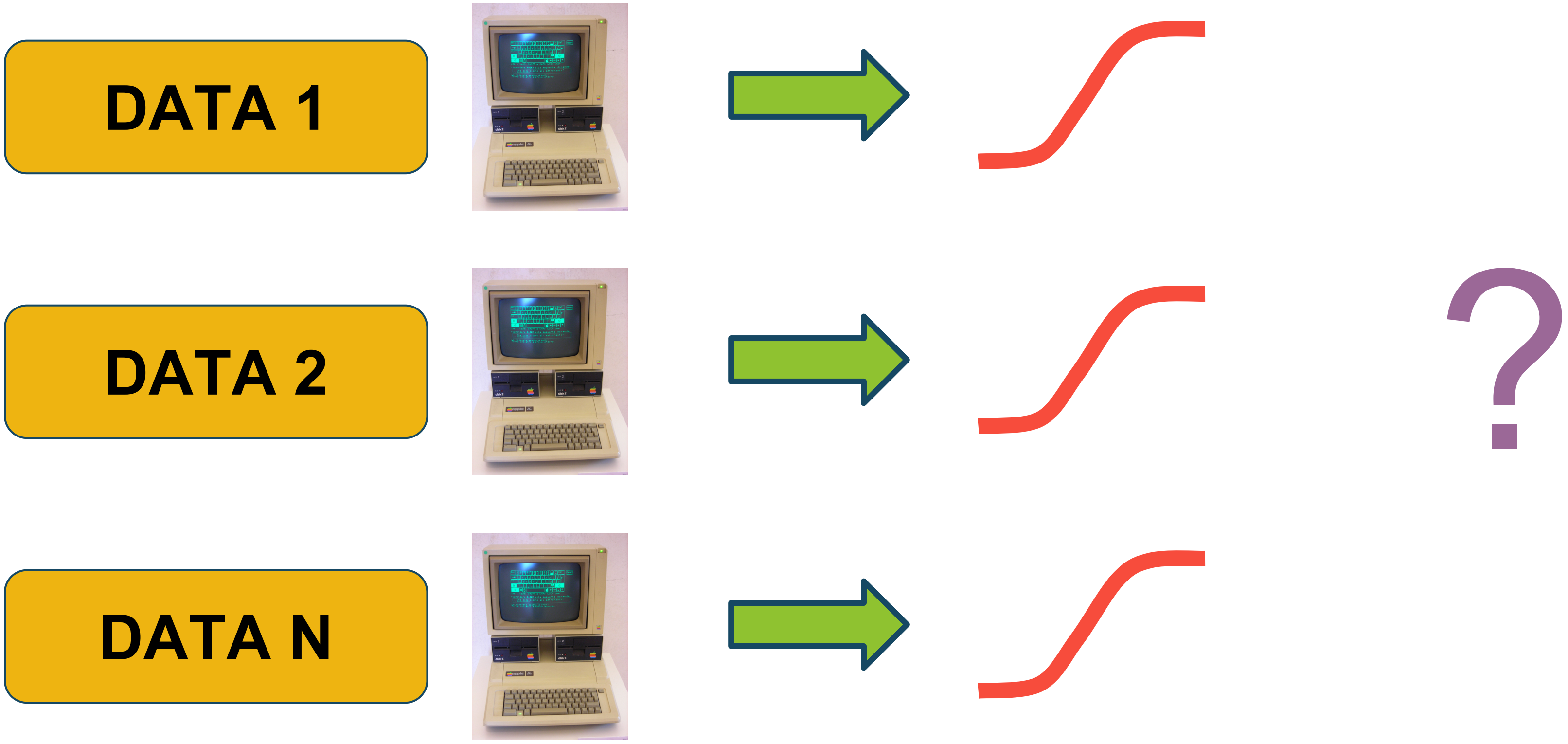
$C_1 \cup C_2$

$C_1 \cup C_2 \Rightarrow$



D_1 |+| D_2

Flashback



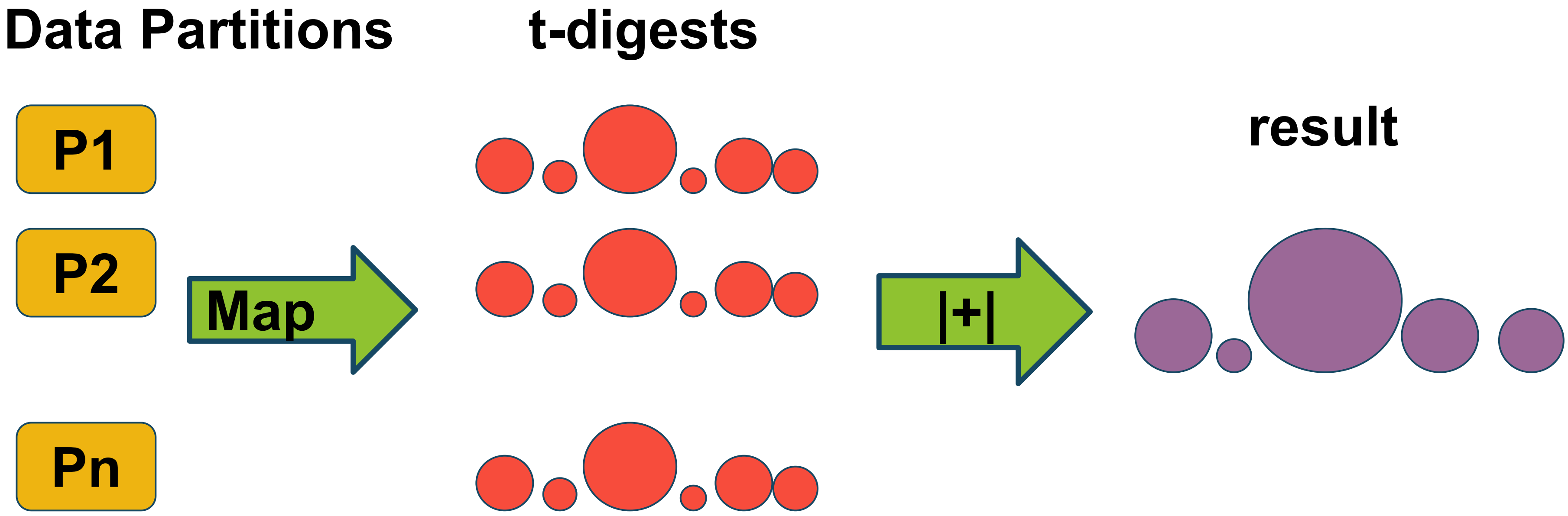
DATA 1

DATA 2

DATA N

?

Mergeable => Scale-Out



Data Partitions

t-digests

result

P1

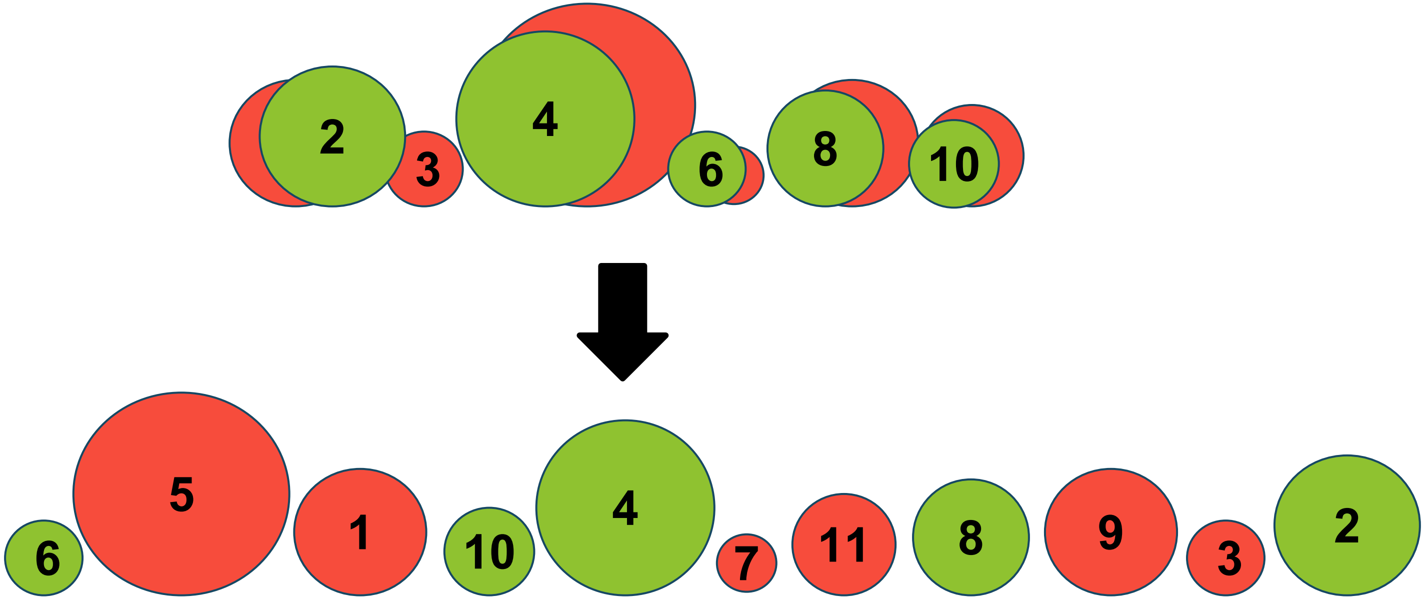
P2

Pn

Map

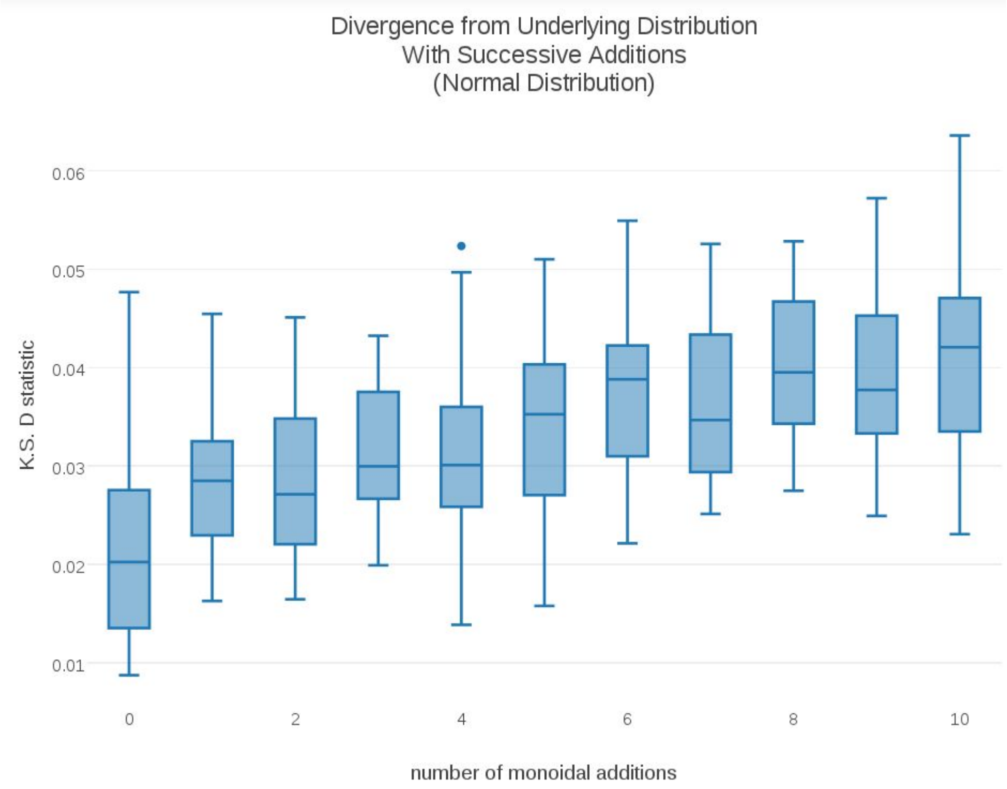
|+|

Merge: Randomized Order

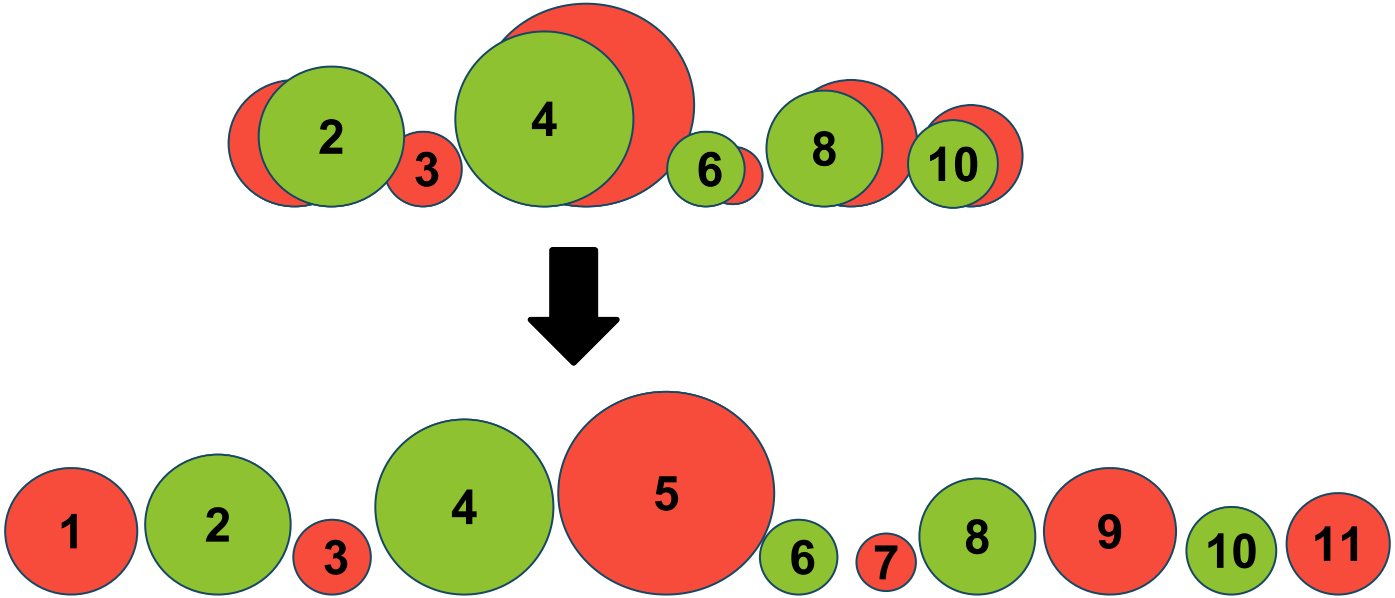


D_1 |+| $D_2 \Leftarrow$

Random Merging Diverges

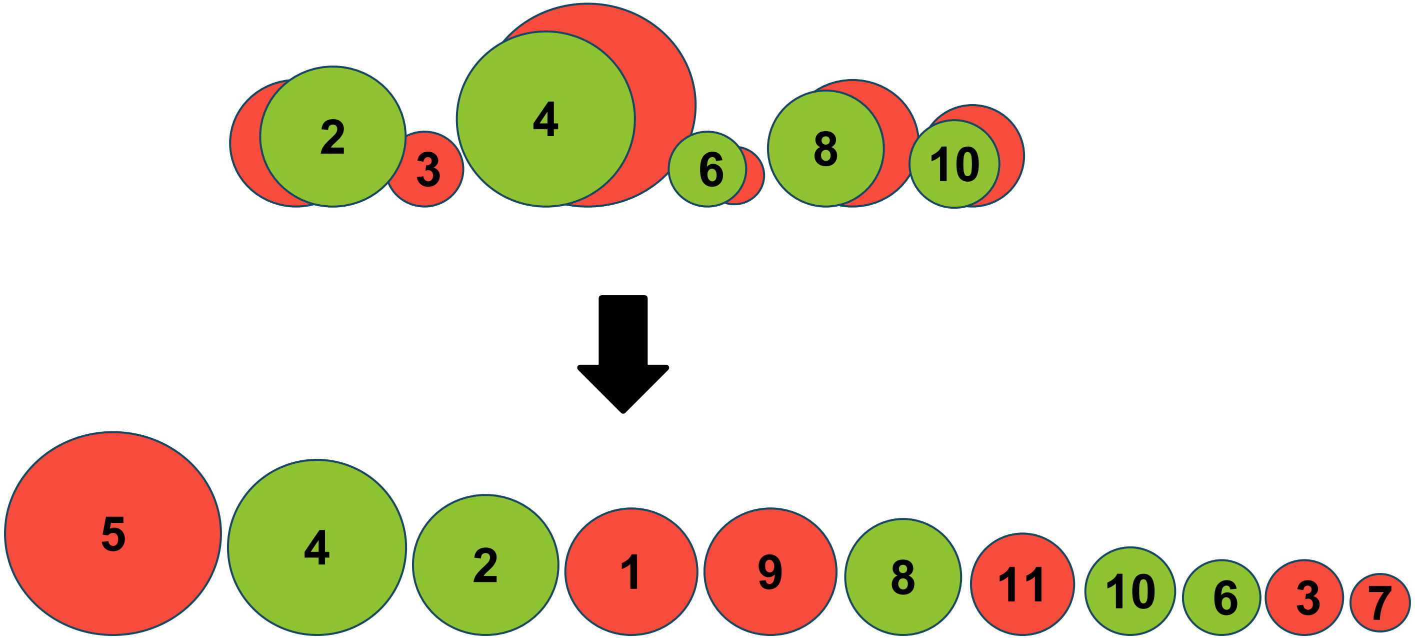


Merge: Location Order



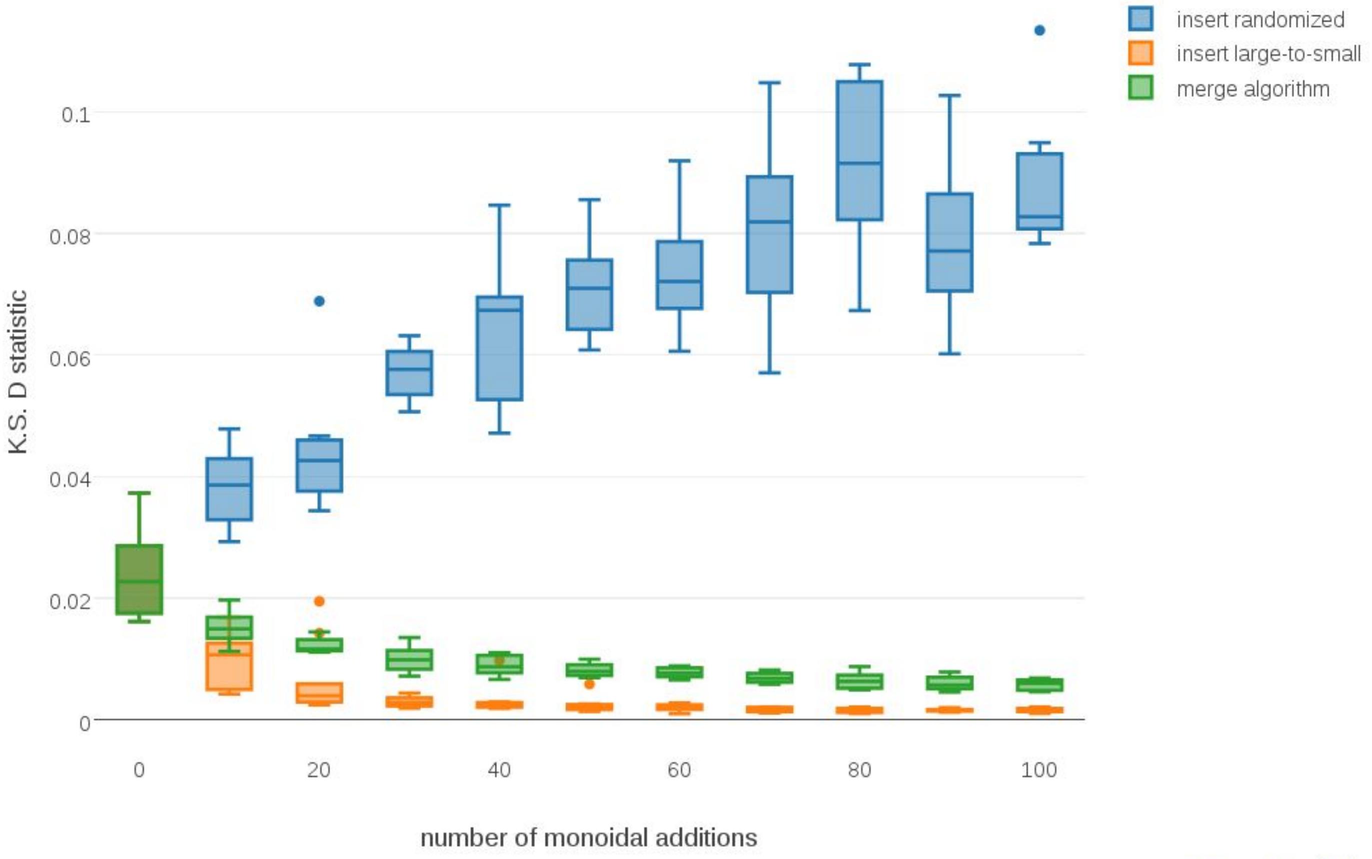
D_1 |+| D_2 ⇐

Merge - Large to Small

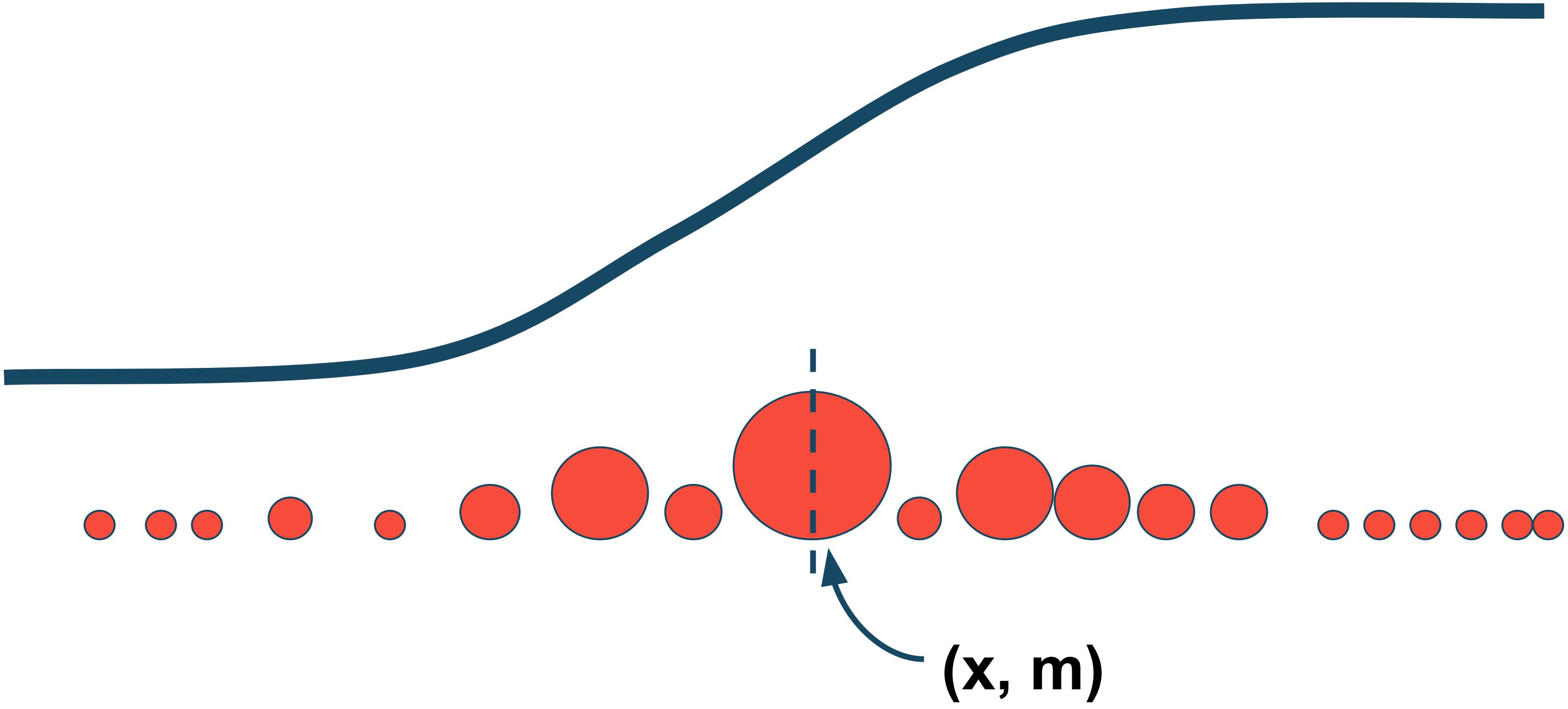


D_1 |+| D_2 ⇐

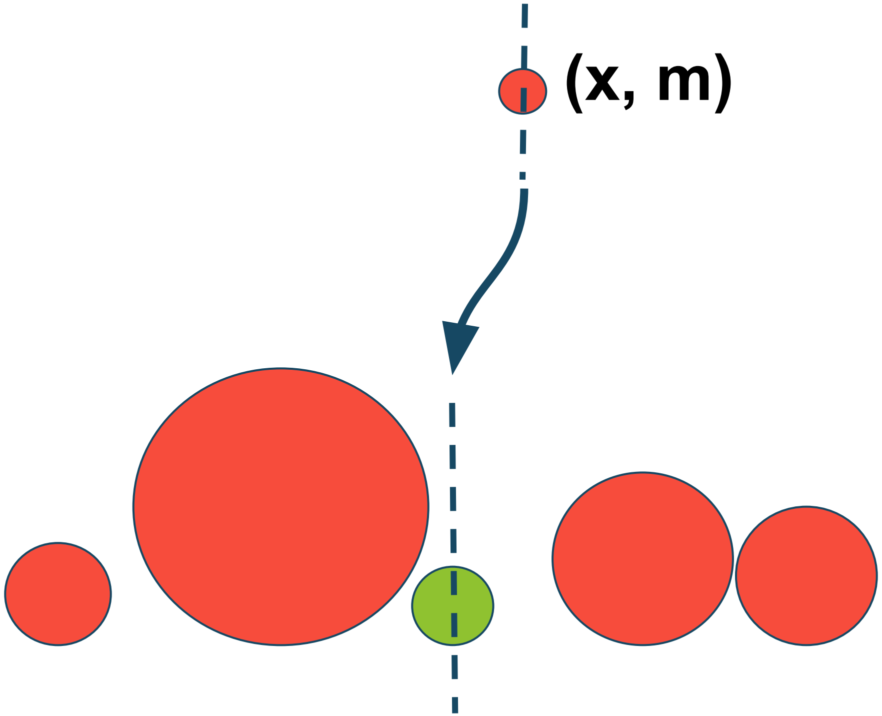
Comparing Merge Definitions



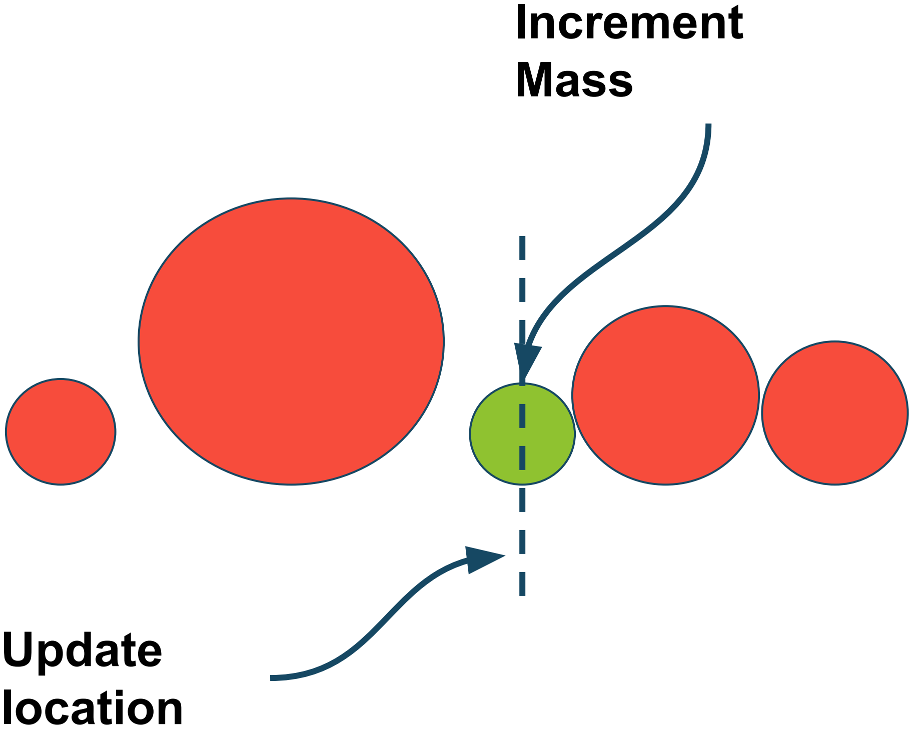
Clusters Maintained in Order



Query the Nearest Cluster

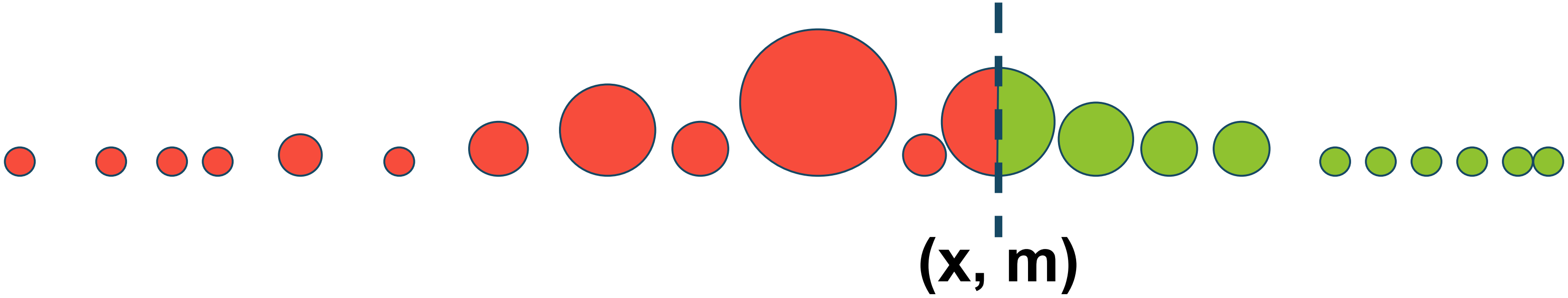


Insert and Update Clusters

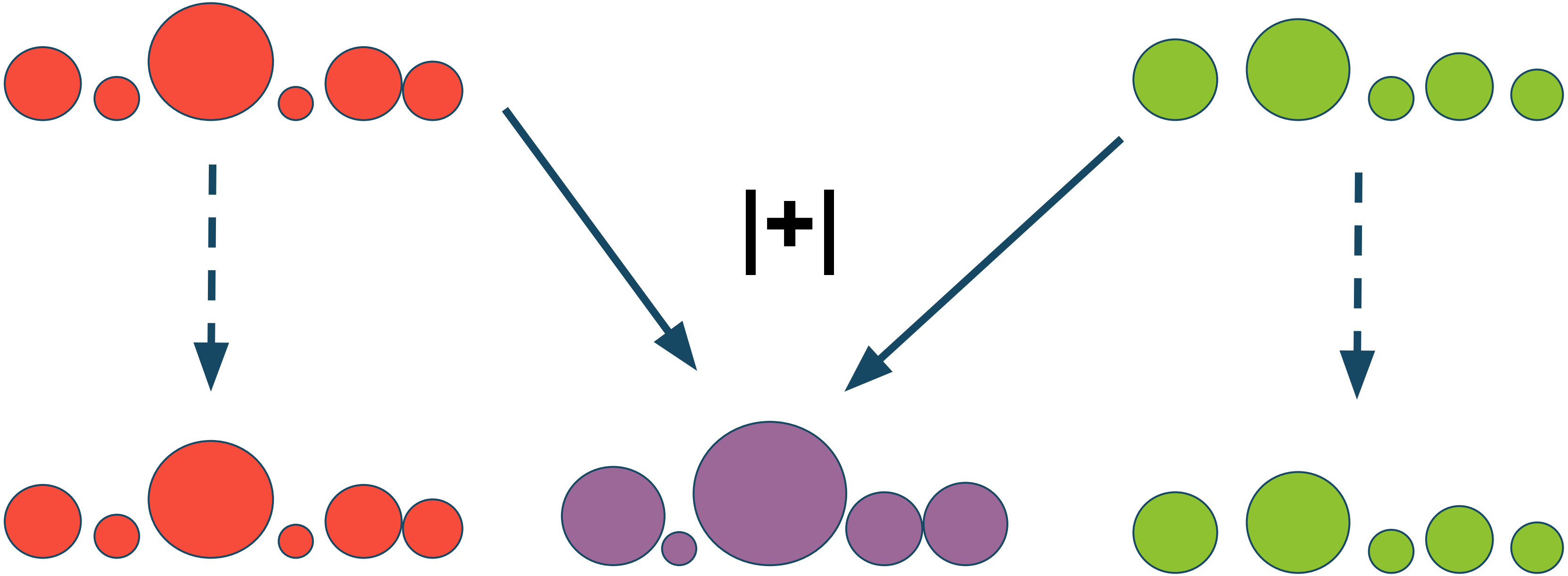


Compute “Prefix Sums”

$$q(x) = \frac{\sum \text{🔴}}{\sum \text{🔴} + \sum \text{🟢}}$$



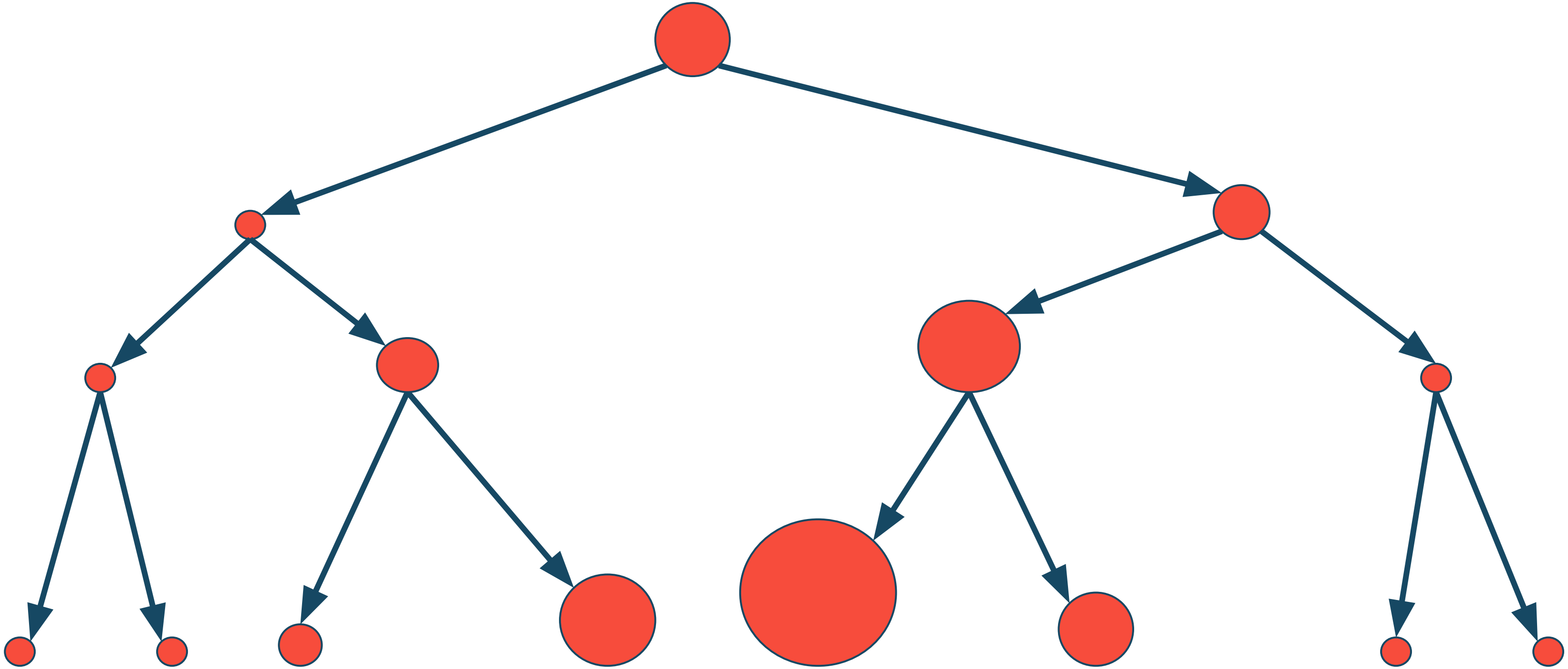
Immutable Data Structure



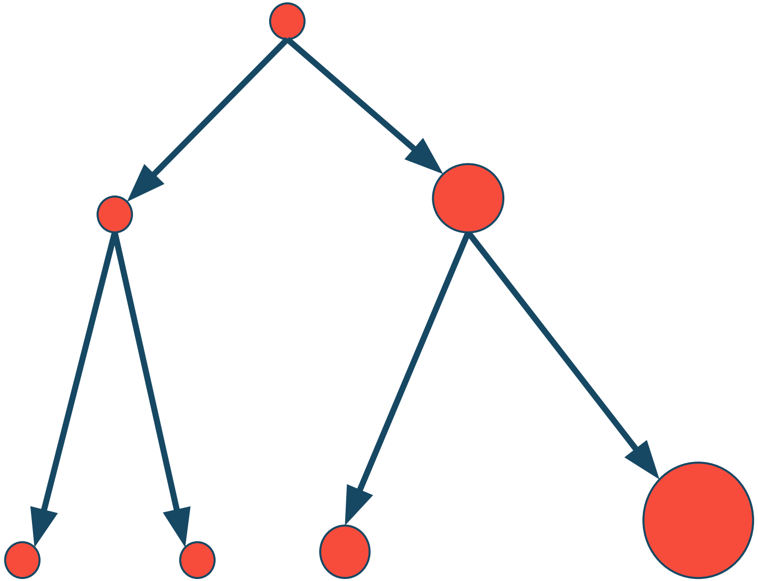
Fast!

$O(\log n)$

Balanced Immutable Tree

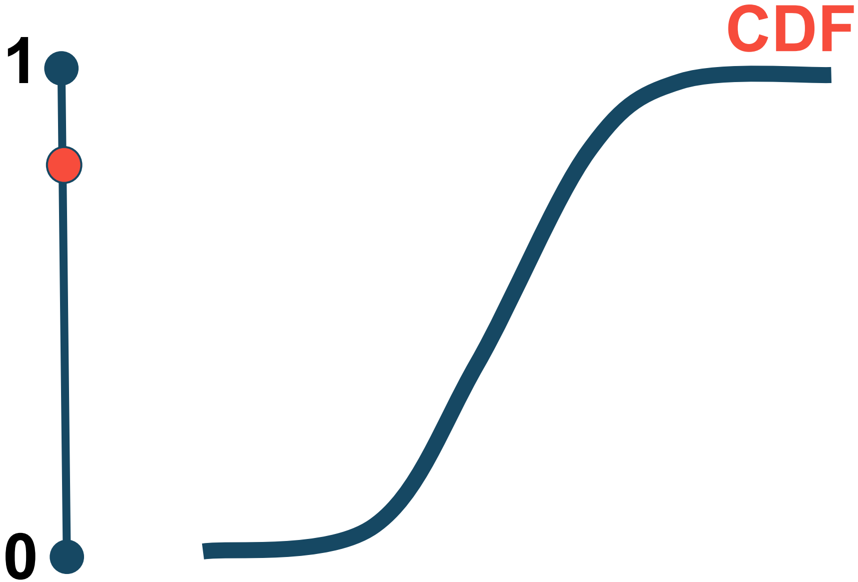


Explore

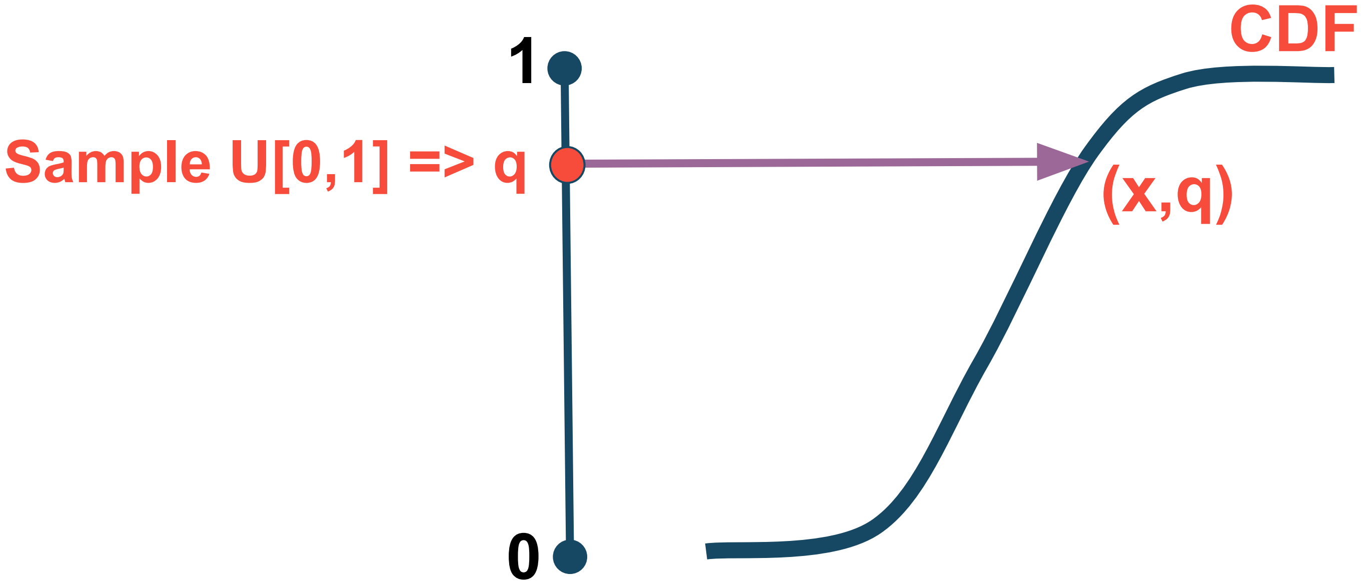


Inverse Transform Sampling (ITS)

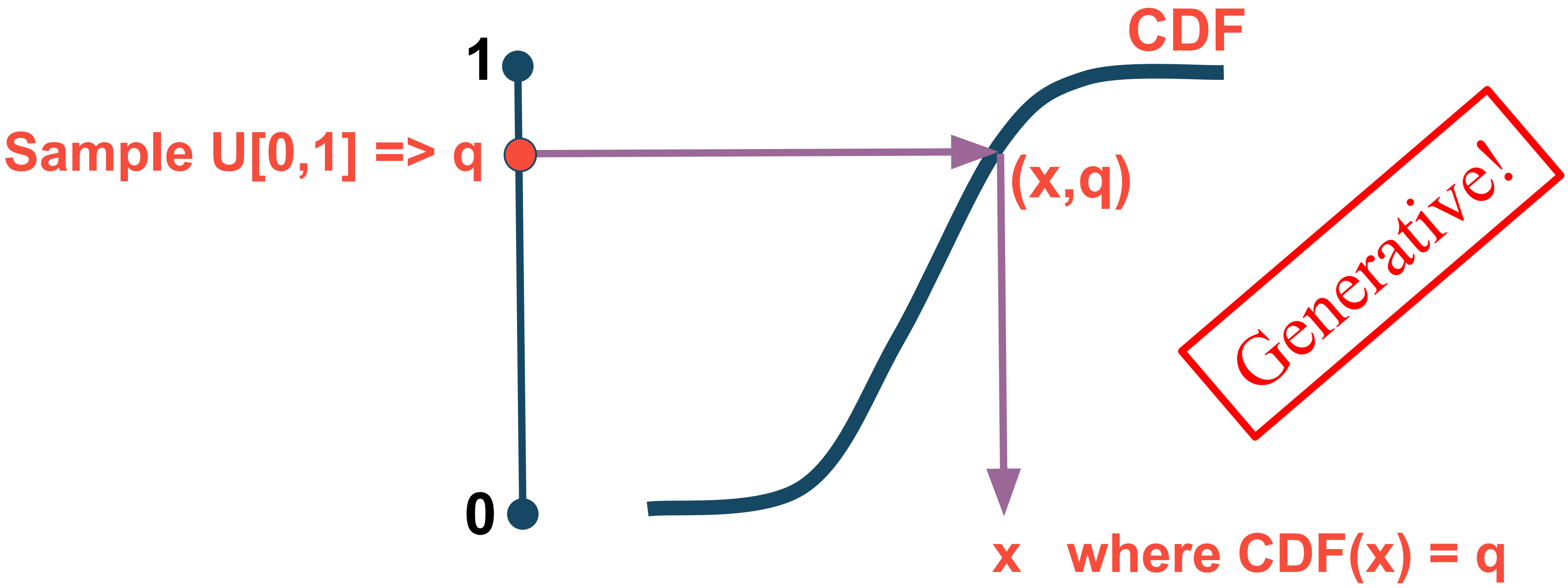
Sample U[0,1] => q



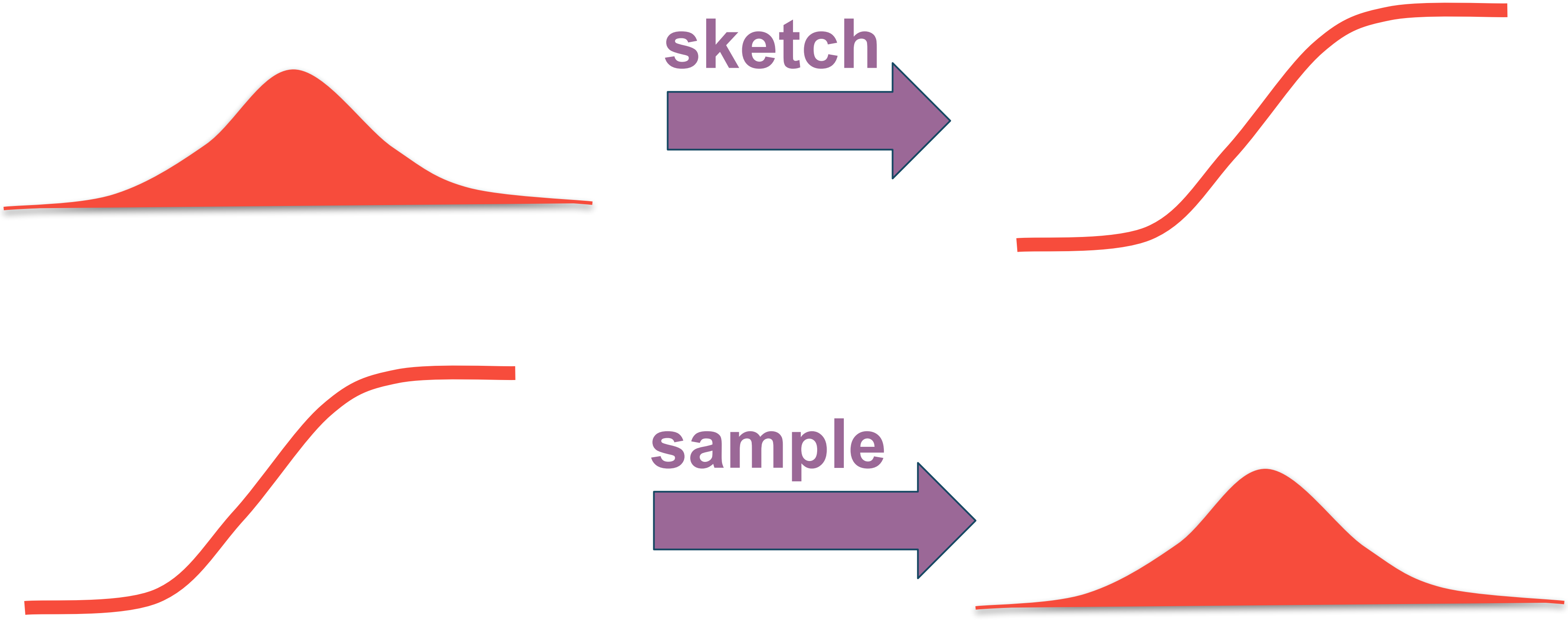
Inverse Transform Sampling (ITS)



Inverse Transform Sampling (ITS)



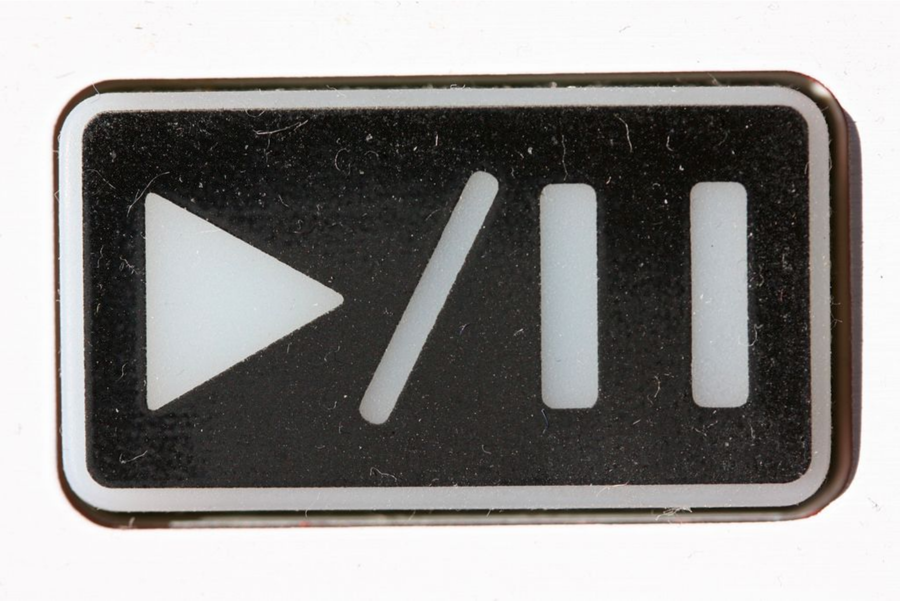
Generative Sampling



sketch

sample

Demo



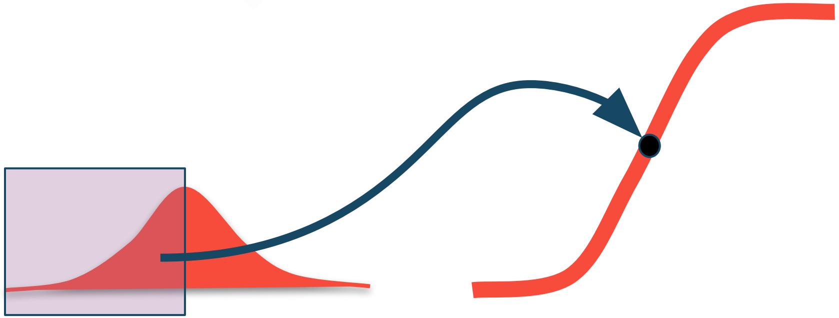
Explore

- T-Digests and Feature Importance for Spark
- Feature Reduction With T-Digests & RF
- Data Science with Generative T-Digests
- Probabilistic Structures for Scalable Computing
- Demo Notebook for This Talk

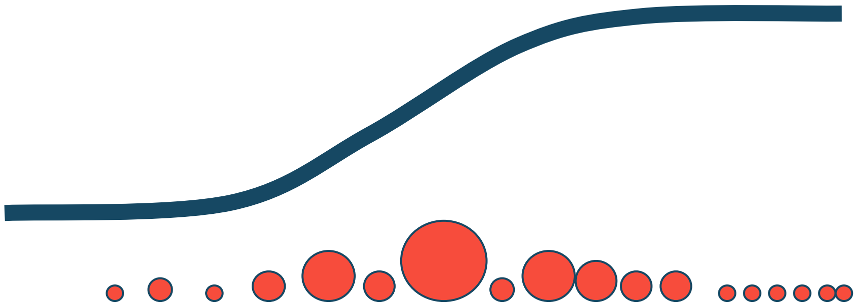
Get This Deck

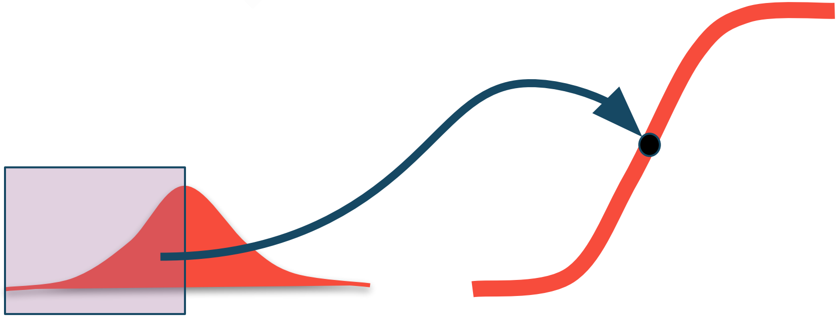
Review

Review

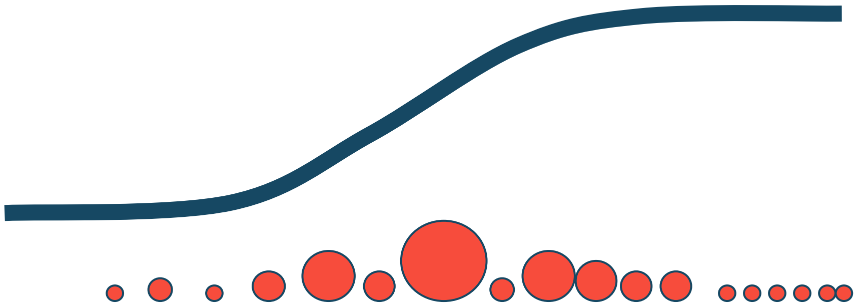


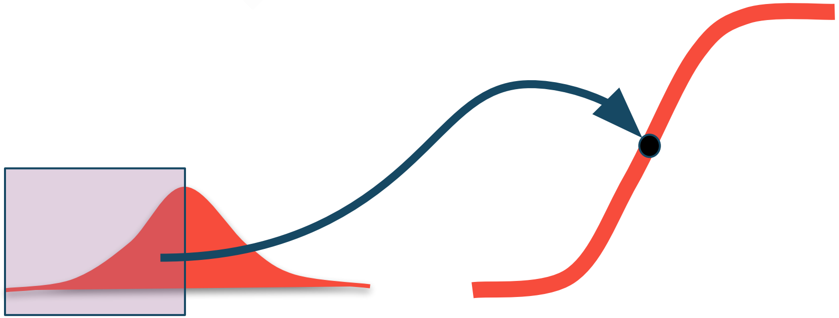
Review

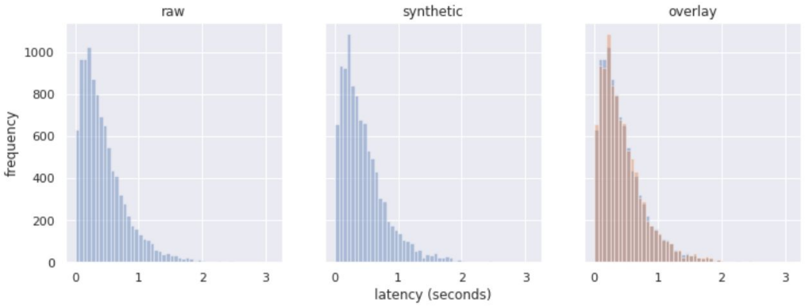




Review






raw
synthetic
overlay
frequency
0
200
400
600
800
1000
0
1
2
3
0
1
2
3
0
1
2
3
latency (seconds)